Can we Teach a Machine to be a Cardiologist?

by

Sonia L. Kleinig

Thesis submitted for the degree of

Bachelor of Engineering (Honours)

in

Electrical and Electronic Engineering

University of Adelaide

2021

Statement of Originality

I declare that all material in this thesis is my own work, except where there is clear acknowledgement and reference to the work of other. I have read the University Policy Statement on Academic Honesty and Assessment Obligations.

This work contains no material that has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published written by another person, except where due reference has been made in the text.

I give permission for this work to be reproduced and submitted to other academic staff for the purposes of assessment and to be copied, submitted and retained by the University's plagiarism detection software provider for the purposes of electronic checking of plagiarism.

______________________ 29 October 2021____

Signed Date

Acknowledgements

I would like to acknowledge my supervisors, Professor Derek Abbott and Mr Mohsen Dorraki, for their constant suggestions and feedback throughout the duration of this project. I would also like to acknowledge my project partner, Hien Long Nguyen, for his efforts and assistance as we cooperated on many aspects of this project.

I would also like to thank all my lecturers, tutors and teachers, both past and present, for helping me get to where I am today.

Finally, on a more personal note, I would like to thank my friends and family for the constant support and encouragement, especially during those final, stressful weeks as my time at university drew to a close.

Abstract

Electrocardiograms (ECGs) are recordings of the electrical activity of the heart, and play an important role in the diagnosis of many cardiac abnormalities. Recently, there has been an interest in finding methods of classifying ECGs using machine learning (ML) techniques. Two major steps are involved with this: pre-processing and classification. Pre-processing techniques, including bandpass filtering and wavelet transforms, are used to reduce noise and extract relevant features from the signal. Then classification is able to be done using a range of techniques. Here, SVM, CNN, and LSTM networks have been used. First, a set of signals with known classification are used to train the ML, and then another set of test signals are used to examine the accuracy of the classifier. This thesis begins with a literature review of previous works in the area of ECG classification with ML. Then, it discusses the method used to apply a range of pre-processing, feature extraction and classification techniques to the same set of data, collected from the PhysioNet Database [2]. The results are examined for an optimal solution. The best results came from the SVM and CNN when wavelet denoising was used to pre-process the data. These achieved F1-scores of 83.6% and 84.8%, respectively. The LSTM network was less effect, only achieving an F1-score of 78.3%, although it is likely this is in part due to the feature set extracted from the data. The SVM and CNN both achieved high results, although future work may be able to improve or build on this. Possibilities include diagnosing between a greater number of cardiac abnormalities, or finding a more optimal combination of pre-processing, feature extraction and classification. Hence, we can conclude that a machine can be taught to be a cardiologist, in the sense that it is able to successfully distinguish cardiac abnormalities when trained appropriately.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation and Significance

Biomedical signals, such as electrocardiograms (ECGs), can reveal unseen details about a patient's health. Hence, it is critical to use these signal to quickly and accurately detect abnormalities to facilitate timely treatment of an abnormal condition.

According to the World Health Organisation (WHO), cardiovascular disease (CVD) continues to be the leading cause of death globally [1]. CVDs can take many forms, and can exist for a number of reasons including behavioural factors, and underlying health conditions [1]. Since most CVDs can be prevented by managing behavioural risk factors, including smoking and unhealthy diet [1], identifying and addressing them early is important.

CVDs generally have some impact on the rhythm of the heart which can be identified from an ECG recording, making them a useful tool in identifying heart diseases. Hence, analysing these quickly and correctly is important, and has led to an interest in using machine learning (ML) techniques to identify heart abnormalities, and even classify the type of abnormality.

This project examines the possibility of using standard ML techniques, programmed in MATLAB, to identify various CVDs. Although this project has no official sponsor, this and similar work could prove invaluable in the biomedical field.

1.2. Project Aims

The aim of this project, *Can we Teach a Machine to be a Cardiologist?*, is to explore various machine learning techniques to determine whether they can be used to teach a machine to correctly classify CVDs. The project involved developing ML algorithms which extracted the relevant features of an ECG, learnt which features correspond to which condition, and then accurately classified other ECG signals according to these features.

An exploration of different techniques should reveal which processes are most able to be used to achieve this goal. The effectiveness of each should be compared with one another, and to results in the literature, with the aim of achieving a comparable outcome.

1.3. Project Scope

This project identifies a number of ML techniques which can be used to classify signals. These techniques will be chosen based on their effectiveness and how easily they can be coded and modified to suit the data available.

This project requires the analysis of ECG recordings. Thankfully, numerous databases with an extensive collection of ECG recordings are available online, so no experiments were needed to collect and accurately label data. The PhysioNet database [2] was used for this project.

1.4. Background Information

This thesis will discuss a number of technical topics. These come under three broad subjects: ECG analysis, pre-processing methods, and machine learning. These topics will be briefly explained here, and further information can be found in the sources referenced.

1.4.1. Electrocardiogram Analysis

In the human body, the contraction of muscles is associated with changes in the membrane potential of cells, i.e. depolarisation [3]. ECGs measure this electrical activity produced by the heart. These measurements are obtained by placing electrodes on the patient's torso or limbs and measuring the electrical activity produced.

Any irregularity in the ECG waveform could be indicative of a CVD or other abnormality. The challenge lies in identifying these abnormalities, particularly since ECG recordings naturally vary from person to person [4], and the abnormalities can be very subtle. Hence, it is important to find a way to accurately identify irregularities and classify the signal accordingly.

Figure 1 shows an idealised ECG signal. A number of points on the ECG are of particular importance. Namely, the P-wave, the QRS complex, the T-wave and the interval between subsequent R-peaks (RR interval).

The P-wave corresponds to the contraction of the two smaller chambers of the heart, the atria. The QRS complex following represents the contraction of the two larger chambers of the heart, the ventricles. This is the contraction that pushes the blood out of the heart and around the body. The T-wave then represents the repolarisation (return to resting state) of the ventricles (note the repolarisation of the atria is hidden in the QRS complex) [3]. Finally, the RR interval represents the time between subsequent heart beats.

Quick analysis of the RR interval can reveal whether or not a patient's heart is beating in a regular rhythm and may point out an arrhythmia if not. Conversely, it is more difficult to determine whether the pattern of P-waves, QRS complex and T-waves are abnormal. This is made yet more difficult by normal variability of ECG features, both within and between patients [4]. Furthermore, electrical activity of other muscles must be taken into consideration when analysing an ECG recording.

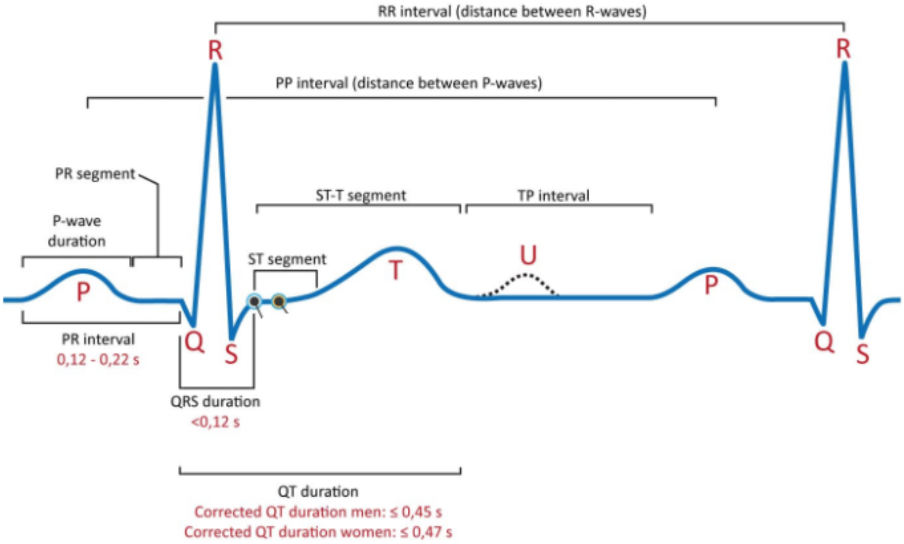


Figure 1: Simplified ECG signal [5]

The database used here (the PhysioNet 2017 Computing in Cardiology Database [2]) contained four types of signals: normal, atrial fibrillation (AF), other arrhythmia, and noisy. Hence, it was important to gain a quick understanding of each.

Normal signals have the characteristic waveform as in Figure 1, although this does have variations from patient to patient. Each feature should have a duration within a specified range, and the RR interval should be fairly constant.

AF is an abnormal condition in which the regular atrial activity has been replaced with fast and disorderly tremor waves [6]. The normal P-waves often disappear, and the distance between R-peaks varies. The incidence of AF increases with age, and can be characterised by palpitations, shortness of breath and chest pain. Figure 2 compares an ECG with AF with a normal ECG rhythm.

Other arrhythmia describes any other abnormal condition. This can include congestive heart failure, a range of blockages, or any other arrhythmia. Noisy signals are classified as signals which contained too much noise to be accurately classified into another class.

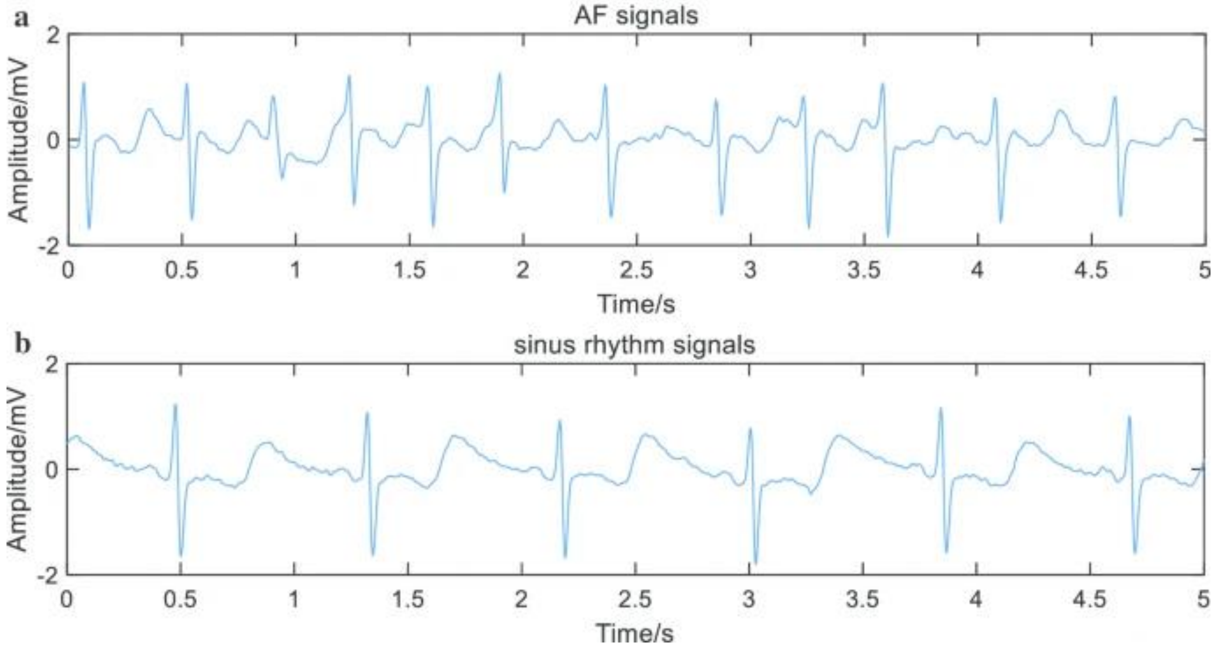


Figure 2: (a) Atrial fibrillation ECG waveform in comparison to (b) normal ECG [6]

1.4.2. Pre-Processing Techniques

Pre-processing of a signal is required prior to analysis to remove artifacts and noise which may impede the classifier. The important features of an ECG are relatively low-frequency (0.5-30 Hz) [6], so much of the high frequency content of the signal is noise which can be removed with filtering [7].

The ECG is unable to distinguish heart activity from other electrical activity in the chest, so other muscle contraction may also be recorded. These are known as artifacts, and may include any muscle movement or slow oscillations from breathing, for example, at the time of ECG recording.

A number of pre-processing techniques exist, including bandpass filtering and wavelet denoising. A brief introduction to wavelets is included here.

Wavelets form an orthonormal basis. This means they can be used to apply a wavelet transform to a time-domain signal to transform it into the wavelet domain in much the same way a set of sinusoids can be used to transform a time-domain signal into the frequency domain, as in the Fourier transform (FT). However, the FT provides only globally averaged information, meaning transient and location-specific features are often lost [3]. Wavelet transforms, on the other hand, allow for time and frequency analysis of a signal simultaneously, which allows transient and intermittent components to be localised.

It is possible to obtain a similar result by utilising a short time Fourier transform (STFT) [8]. This computes the FT of a signal in smaller time windows and can be used to plot an image known as a spectrogram (see Figure 3). However, wavelet transforms do a better job of this since they apply a window of varying length to the signal [3,8]. This allows the transform to adapt based on the frequency components of the signal, which is much more difficult to do with FTs. The time-frequency plot produced by the wavelet transform is called a scalogram, and an example is shown in Figure 4. Notice that it is similar to the STFT spectrogram, but provides more information, particularly at higher frequencies.

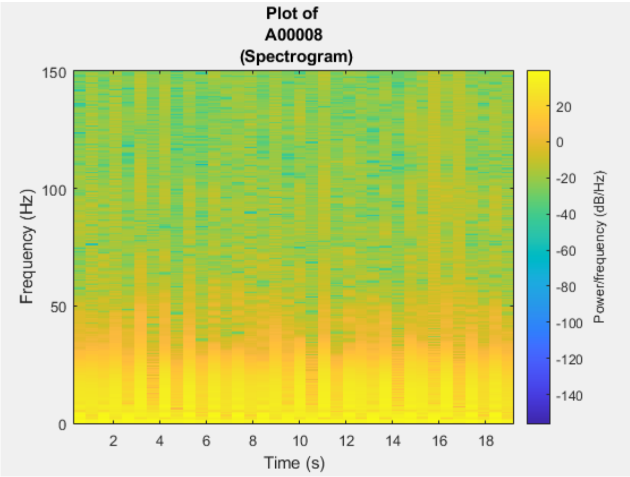


Figure 3: Example spectrogram

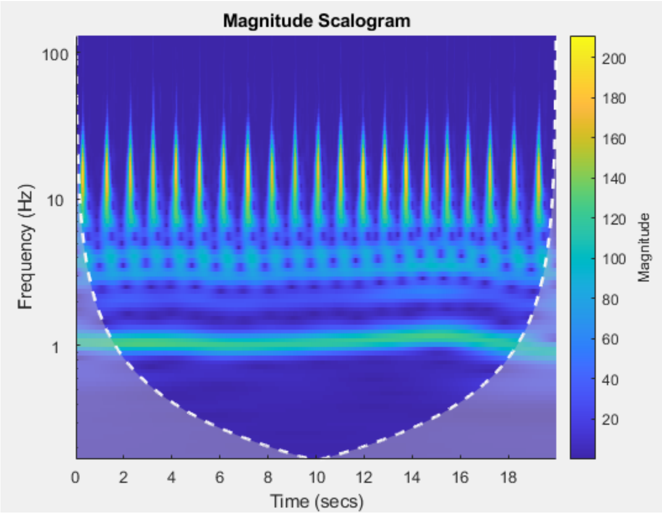


Figure 4: Example scalogram

Furthermore, a number of different wavelets may be used. The most popular of these are shown in Figure 5. The type of wavelet used for a given application can be chosen to best match the signal being analysed. Notice that some of these wavelets have a similar shape to the ECG waveform, and prove can be useful in this case.

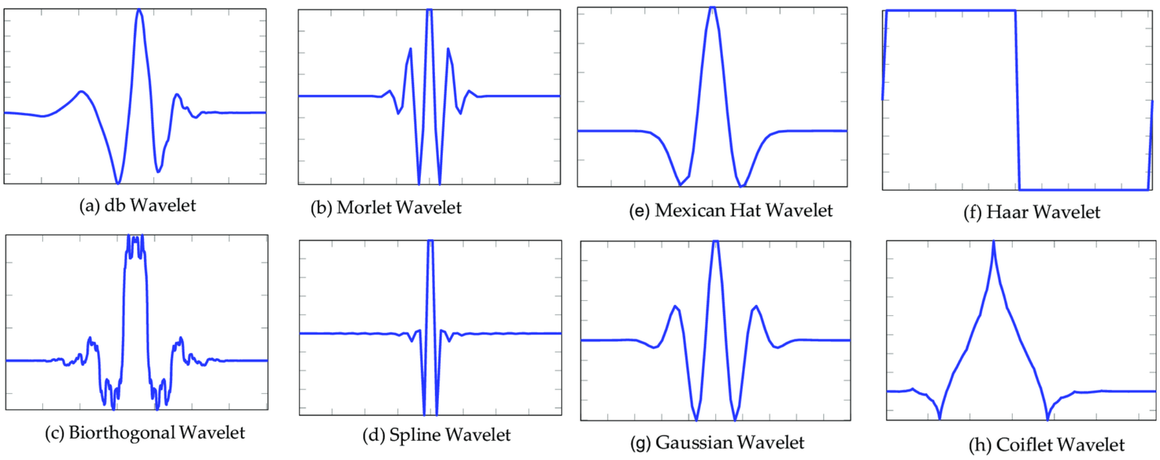


Figure 5: Common wavelets [9]

1.4.3. Machine Learning Techniques

ML is an application of artificial intelligence in which algorithms parse data, learn from it, and apply what they've learned to make an informed decision [10].

First, a few definitions are important to understand. Before it can be used as a classifier, ML algorithms must be trained in how they should classify that data. As such, a set of data must be divided into a 'training set' and a 'test set'. The training set and the correct label for each signal is given to the machine to teach it about each classification. Then the test set is used to verify how well the machine has learnt these classifications. The machine's classifications can be compared to the actual label for each data to calculate the accuracy and other metrics for the algorithm.

The following ML techniques have been identified:

- Support vector machine (SVM);
- Artificial neural network (ANN);
- Convolutional neural network (CNN); and,
- Long Short-Term Memory (LSTM) networks.

A basic description of each of these techniques is included here for readers who have not encountered these terms before.

An SVM is a supervised machine learning algorithm which can be used to assign labels to data, based on the value of a number of features it possesses. Each data item is plotted in n-dimensional space, where 'n' is the number of features under consideration [11]. Then, the SVM draws a line, or a hyperplane in higher-order space, which best separates the data in the training set into its known categories. The test set is then plotted in this n-dimensional space, and classified according to which side of the hyperplane it falls on. Figure 6 illustrates a simple 2D example of this concept, in which the solid line shows the plane between classes (red and blue). In this case, if the next test sample, shown in green, was added to the data, it would be inferred as a Class 2 item, due to the side of the hyperplane it falls on.

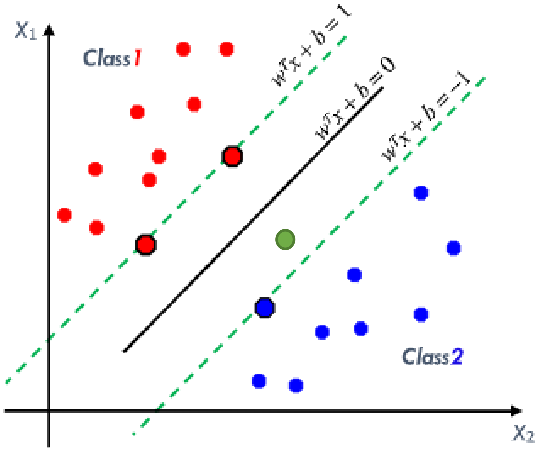


Figure 6: 2-dimensional example SVM [12]

An ANN is capable of extracting complex and non-linear relationships between features of a set of data [12]. They are constructed to simulate neurons in a biological nervous system, as depicted in Figure 7. It's comprised of many interconnected units, whereby the network function is largely determined by the connections, and each connection is a certain nonlinear function. The weight of each connection determines its contribution, and these weights can be adjusted through training, either from outside information or in response to the inputs [13]. The network is built directly form experimental data and the ANN's self-organising capabilities, and does not require prior assumptions [13].

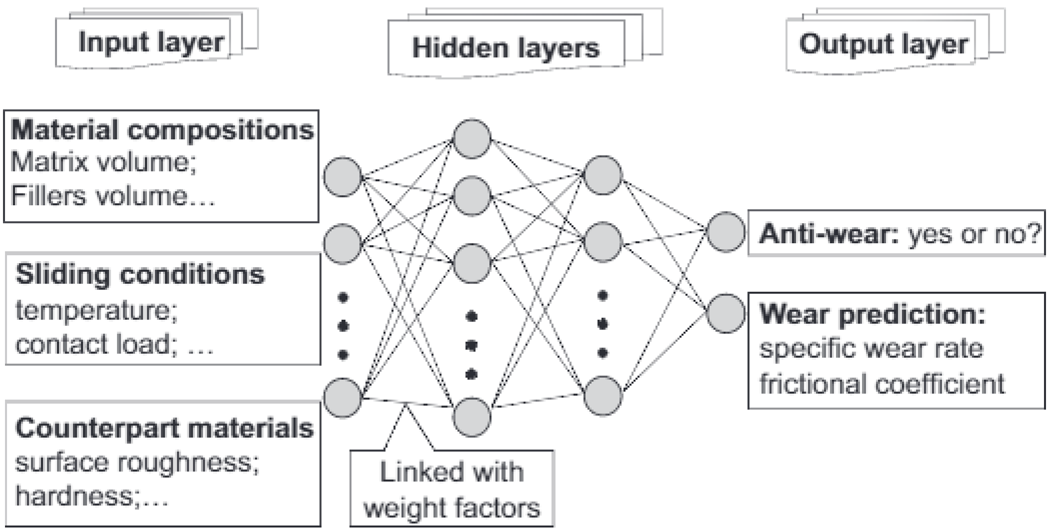


Figure 7: Example of an ANN for correlating certain properties with various parameters [13]

Building on from ANNs, Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) well-suited to time-series data [14]. They are an improvement over traditional RNNs which suffer from short-term memory, meaning they have a tendency to "forget" earlier information. An LSTM network has the ability to keep or forget information as training progresses [15], meaning they can learn to consider only the important features during classification. This enables it to effectively analyse long sequences of data. Figure 8 shows the structure of an LSTM unit.

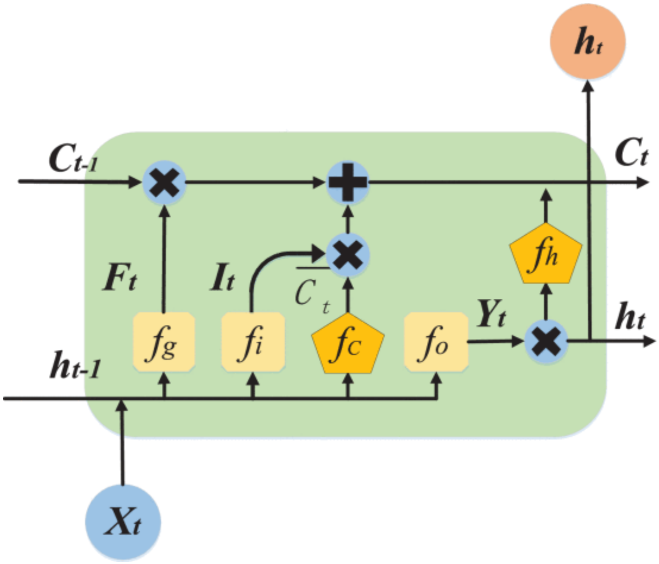


Figure 8: LSTM unit structure [15]

Alternatively, CNNs add some processing stages to the input of the neural network. They are especially helpful for classifying images, such as handwritten symbols as shown in Figure 9. The convolution layer extracts features, and the pooling layer reduces the size of these features to decrease computational power [16]. These two layers enable the model to understand the features. Multiple convolution and pooling layers can be used to extract higher-level features than possible with a single layer. Finally, a fully-connected layer is used to classify the images, and this is generally a regular ANN [16].

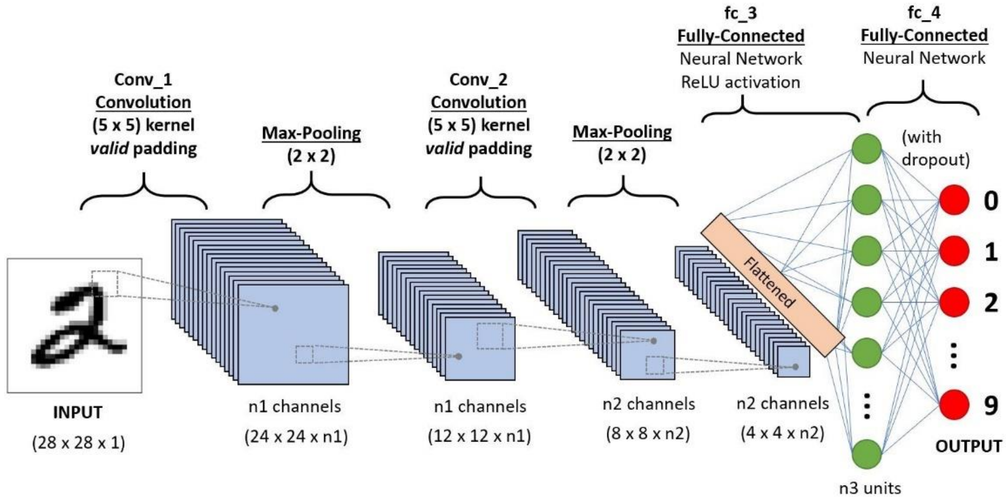


Figure 9: CNN example [16]

Neural networks come under the category of *deep learning*. This is a subfield of machine learning in which algorithms are structured in layers to create the neural network. This makes it possible for the machine to learn through its own method of computing [16]. Conversely, basic ML requires human guidance to improve the process of its classifications, such as with the SVM.

1.5. Technical Challenges

A number of technical challenges were identified. Perhaps the biggest, is the use of programming software. The project team has only a basic understanding of MATLAB, and will need to develop this knowledge in order to be able to complete the project. A possibly more effective alternative is to use Python, however, neither member had any previous experience with this language. An attempt was made to learn this, but due to time constraints, it was decided to just use MATLAB.

The project involved finding ML algorithms which could successfully classify ECG recordings, although neither member has previously worked in ML. This challenge was mitigated by beginning research into ML concepts early, and by following coded MATLAB examples to get a deeper understanding of how these techniques work.

Finally, ML algorithms can become quite complex to run, and as such the computational power and time available may present a challenge to this project. For example, a reasonably powerful computer (generally more than a laptop) is required to have enough processing power and RAM to run more complex algorithms. The MATLAB Parallel Processing toolbox was needed to allow the GPU to be used, which significantly decreased ML training times. Where this wasn’t enough, alternatives such as using Adapt through the University was considered as a viable alternative.

2. Literature Review

2.1. Introduction

From preliminary research, it was decided that the project would be split into three main objectives, namely pre-processing, feature extraction, and classification. Hence, the literature review has a similar structure. The focus was directed almost exclusively at papers which applied these processes to ECG analysis.

The literature review covers the following topics:

a. Suitable pre-processing techniques;

b. ECG features and feature extraction techniques;

c. The effectiveness of various classification methods, including SVM, CNN and LSTM classifiers; and,

d. A comparison of various combinations of pre-processing, feature extraction and classification techniques.

The findings for each of these points are discussed in the subsections of Section 2.2, and conclusions are drawn in Section 2.3.

2.2. Findings

The findings of the literature review can be divided into the three stages of pre-processing, feature extraction, and classification. Although each stage will be discussed individually, it should be noted that most papers use one or more techniques in each stage. The effectiveness of each classification method is summarised in Table 1.

2.2.1. Pre-Processing Methods

Pre-processing is an important first step in classifying the ECG signals [7,17,18]. It removes undesirable features including noise, baseline wander, motion artifacts and other interruptions [17].

2.2.1.1. Noise Removal

Noise removal can be achieved in a number of different ways. It can be as simple as using a bandpass filter, as in [6,19,20]. Wang et al. [20] used a number of different filters to pre-process the ECG recordings. These were a 50 Hz notch filter to remove the mains power hum, a 30 Hz low-pass filter to remove high-frequency noise, and a 0.1 Hz Chebyshev high-pass filter to remove low-frequency noise and artifacts.

Hu et al. [6] used a similar filtering process, with a digital FIR bandpass filter which had cut-off frequencies at 0.5 Hz and 30 Hz. The 30 Hz cut-off frequency could eliminate some electrical activity from muscle movement, as well as powerline interference. The 0.5 Hz cut-off was chosen to remove low-frequency artefacts due to respiration, electrode movement, and other low-frequency noise [6].

Alternatively, wavelet denoising can be used. The wavelet decomposition of a noisy signal concentrates the signal information across a few wavelet coefficients, without modifying the random distribution of the noise [9]. This meant denoising could be achieved by thresholding the wavelet coefficients. Another advantage of the WT is that it gives a time-variant decomposition, making it possible to choose different filtering settings for different time windows [9].

Adaptive filtering methods can be developed from Least Mean Square (LMS) and modified LMS-type algorithms [21]. The concept of these algorithms is to estimate signals which are corrupted with additive noise by minimising the mean squared error between the noisy ECG signal and a reference signal containing noise correlated to that in the ECG [21]. Venkatesan et al. [17] proposed a delayed error normalised LMS, which they found to be superior to both normalised LMS and delayed normalised LMS. Similarly, Khiter et al. [21] proposed a self-correcting leaky normalised LMS (SC-LNLMS), which applied multiple iterations of an adaptive noise canceller to further reduce signal noise.

2.2.1.2. Other Pre-Processing Steps

Pre-processing can also include other steps, such as segmenting ECG signals [6,17,20,22,23,24,25], and heartbeat normalisation [24]. The length of signal segment can vary, with some methods using long signals (i.e. >30 seconds) [17,25], and some used 10 second intervals [20,22,23,24], or even less [6].

Multiple pre-processing steps could be combined to create a more robust pre-processing method. Zhao et al. [24] used an FIR highpass filter to remove baseline drift and other low frequency noise. The signal was then decomposed into three levels by a biorthogonal spline wavelet, and a threshold was selected to remove the unwanted frequency components. The heartbeat of each signal was normalised to 75 bpm to remove the effect of natural variations in heartrate. Finally, the quality of the signal was measured [24], to verify only clean signals were passed to the next stage of the classifier.

2.2.2. Feature Extraction

Feature extraction involves the analysis of raw data to extract relevant features. These features are used to classify a signal into a class with similar features, and distinguish it from classes with different features. For ECG signals, a number of different features are of interest. These may exist as either time-domain or frequency-domain features, or even time-frequency features.

2.2.2.1. Time-Domain Feature Extraction

The detection of R peaks and/or RR intervals was a popular time domain feature [6] [19,26,27]. A number of papers also detected the whole QRS complex [23,28], and some studies extended this to more features including the QT-interval and P-wave duration [23]. The time between subsequent R-peaks corresponds to one heartbeat, so if the time between R-peaks is inconsistent, this could indicate an abnormality [6].

Wang et al. [26] extracted the RR intervals (time between subsequent R-peaks) from their ECG data. They then compared subsequent RR intervals and the ratio between these. This data was one of two sets of data used to train and test their CNN (the other being spectrograms). Hu et al. [6] also examined the difference in RR intervals, although just for AF and normal ECGs. Their results showed that AF signals had an RR interval half the length of a normal signal, and there was greater variance between the RR intervals.

Bae et al. [27] discussed how a pair of derivative filters could be used to detect R-peaks and the QRS complex. They also demonstrated a noise detection algorithm which could be used to exclude contaminated R-peaks. It was suggested that developing a technique to measure the reliability of detected R-peaks and calculated RR intervals may be just as important as developing a QRS algorithm with a higher detection rate [27]. The application of this research was for real-time detection of arrhythmias, in which case it was especially important to remove the unreliable RR intervals [27].

Numerous papers report on the use of a K-Nearest Neighbour (KNN) algorithm to detect R-peaks [19,23,28]. KNN is a type of instance-based learning in which an object is assigned to the class of its k nearest neighbours, where k is an integer. For the case where $k = 1$, the object is assigned the same class as its single nearest neighbour [19].

Using KNN to detect either R-peaks or the QRS-complex has proven successful. He et al. [19] achieved a peak detection accuracy of 99.43%, Saini et al. [28] achieved QRS detection rates of up to 99.89%, and in a different paper [23], accomplished results of 100% for detecting QRS duration, heart rate, QT-interval, P-wave duration and PR-intervals.

A method called the Pan-Tompkins algorithm has also been used to identify the QRS complex from ECG signals [4,22,29]. In this method, the signal is filtered and differentiated to remove noise and suppress the lower frequency components of the P and T-waves. Squaring further enhances the high frequency components, and a moving window integration extracts the slope of the R-wave [4]. Further threshold adjustment may be done to improve sensitivity.

Saadatnejad et al. [29] used the Pan-Tompkins algorithm to extract a number of time-domain features from their ECG data prior to classification. This included the time interval of the past and next RR intervals, the average of the previous and next five RR intervals, and the average duration of the RR intervals in each patients' data. They argued that other time-domain features such as those based on the QRS complex, the R-wave or the T-wave did not efficiently represent the differences among the arrythmia classes examined [29].

Finally, it is also possible to detect the RR intervals using wavelets. For example, Venkatesan et al. [17] uses a Coiflet wavelet to detect the RR interval and possible R-peaks. Although wavelets are commonly used to extract features, they are mostly used to extract frequency or time-frequency information instead [4,8,26,30].

2.2.2.2. Frequency-Domain Feature Extraction

The main features of an ECG signal are contained within the frequencies of about 0.5 Hz to 30 Hz [6,20], with components outside of this range largely corresponding to noise.

Hu et al. [6] demonstrated that the maximum amplitude frequency component of an ECG may be an important feature. The maximum amplitude frequency component was consistently close to 1 Hz for normal signals, and much more volatile (ranging from 2 to 8 Hz) for ECG recordings with AF [6]. This study did only analyse two conditions, with no mention of whether or not this feature can be generalised to other abnormal conditions.

ECG signals are non-stationary data, and as such, their instantaneous frequency changes with time [18]. This means their properties can't be fully described just by using frequency-domain information [18]. Therefore, a means of combining this frequency information with time-domain information is required.

2.2.2.3. Time-Frequency Analysis

One simple time-frequency metric is the instantaneous frequency (IF) of a signal. The IF is defined as the mean of the frequencies in the signal as it develops [31], or the frequency of a sine function that locally fits the signal [32]. Appropriate features are then extracted from the IF signal, often in the form of statistical features. However, IF is sensitive to the amplitudes of the frequency components, and this non-linear nature severely constrains the spectral interpretation of the IF [32]. A possible solution to this is to instead use the Moment of Velocity (MoV). The MoV can suppress larges spikes that often clutter the IF, even in the presence of noise [33]. It has been shown to be more effective at identifying the location of R-peaks in the ECG waveform [32], when compared to IF.

Another option is to convert the 1D signals into a 2D image. Spectrograms and scalograms are powerful tools in the analysis of ECG recordings [3,18,26,30]. They demonstrate how the frequency content of a non-stationary signal varies with time (refer back to Section 1.4.2 for further background information). Spectrograms and scalograms are important for this application as they can be saved as images for input to classifiers, such as the CNN [18,26].

Huang et al. [18] plotted spectrograms as 256×256-pixel images before using them to successfully train and test a CNN. These images were created using a STFT over 10 seconds of a recorded signal. Rashed-Al-Mahfuz et al. [30] produced scalograms of segments of ECG signal as input to a VGG16-based CNN. The results were compared to ones obtained using a Hilbert-Huang Transform (HHT), in which case the scalogram overperformed the HHT in all cases. Wang et al. [26] also used the WT to produce scalograms of ECG signals, to use as one input to their CNN (along with RR interval information).

Even without plotting a spectrogram, the WT can be used to decompose a signal into a series of wavelet coefficients [29,34]. Emanet [34] converted each signal into 265 wavelet coefficients, and MathWorks [25] selected 190 features. The idea behind this is to represent the signals as a smaller number of data points, and for the case of [25] this meant decreasing the points down from 65536-length vectors to just 190. Similarly, Saadatnejad et al. [29] employed a Debauches-2 wavelet decomposition into 4 levels for their analysis.

2.2.3. Classification with Machine Learning

Machine learning techniques can be used to classify ECG signals, based on a number of learned features. Preliminary research identified a number of possible classification methods, namely:

a. SVMs [17,25,35,36];

b. CNNs [18,26,30,37,38];

c. RNNs with LSTM [29,39,40];

d. k-Nearest neighbour algorithm [24,41,42];

e. Random Forest algorithm [34,36,43,44]; and

f. Decision trees [6].

This review focusses on the two most relevant, those being SVMs and neural networks (including LSTM and CNN), with the other techniques being discussed more briefly.

2.2.3.1. Support Vector Machine

The SVM is a widely adopted pattern recognition, object identification, and image classification technique [17]. Venkatesan et al. [17] used an SVM classifier to sort ECG recordings into a normal and abnormal set based on a range of time-domain and frequency-domain features. This achieved an accuracy of 96% [17]. An example MATLAB program written by MathWorks [25] also obtained an accuracy of 97.96% with an SVM.

Zhang et al. [36] tested a couple different SVMs, including a kernel SVM (KSVM) and least-squares SVM (LS-SVM). Their results found the traditional KSVM to have the worst results, and the LS-SVM to be the most effective of the methods compared, with an accuracy of over 92%.

Li et al. [35] extended the idea of the SVM by experimenting with different ways in which it could be optimised. Particle swarm algorithm (PSO), genetic algorithm (GA) and the grid search algorithm (GS) were each used to optimise the SVM which was used to classify between six ECG beat types. The results for each were high, with average sensitivities (SE), specificities (SP), positive predictive values (PPVs) and accuracies each well above 95% [35] (see Table 1 for specific values).

2.2.3.2. Convolutional Neural Networks

CNNs are a type of deep learning model commonly used in image and data analysis, as well as disease classification [38]. See section 1.4.3 for further background information.

Huang et al. [18] reported an average accuracy of 99.00% with their 2D-CNN. This classifier used three layers of convolution and pooling. For comparison, they demonstrated a 1D-CNN which produced an accuracy of 90% when supplied with a similar sized test set. Wang et al. [26] also used a CNN with three convolution and pooling layers. Although they produced a high accuracy of 98.74%, the PPV, SE and F1-scores of their method were lower, with 70.75%, 67.47%, and 68.76%, respectively [26].

Rashed-Al-Mahfuz et al. [30] used a *VGG16* architecture, which consists of a CNN with 16 layers in order to classify an input image. They found accuracy was improved when a CWT scalogram was used instead of HHT spectrum. The accuracy of this classifier was also dependent on the number of classes to be distinguished between. When two, three or four classes were being distinguished the classifier could achieve a 100% accuracy, but had a lower accuracy of 99.9% when six classes were used [30].

Dokur et al. [37] considered the accuracy of both 1D-CNN classification, and the classification of ECG images using CNN which had been trained with Walsh functions. Walsh functions form an orthonormal basis like trigonometric functions in Fourier analysis, but have a number of advantages including the ability to easily expand the number of learned classes [37]. The results found this method achieved an accuracy of 99.45% for the classification of 1D ECG signals, and a 98.7% accuracy for 2D ECG images.

Lih et al. [38] made use of a model called Long Short-Term Memory (LSTM) to improve the results obtained from their CNN. Although the training of this approach was time-consuming and required a sizeable amount of data, the system was able to achieve a high classification accuracy (97.33%) despite using signals with noise [38].

It was recommended that a pre-trained model with high-performance in a related task be used to reduce computational complexity [30]. Parts of the classifier can then be adapted as needed to tailor its performance.

2.2.3.3. Recurrent Neural Networks with Long-Short Term Memory

LSTM networks are a widely-used type of RNN, designed for time-series modelling [39]. They overcome the problem of "gradient vanish or explode" which occurs in traditional RNNs when dependencies are too long. Instead, key features of the data can be efficiently maintained in the LSTM by a number of gate units and memory cells in its structure. Of especial note, is the forget gate, which allows the network to forget irrelevant information [39].

Saadatnejad et al. [29] used two RNN models, which were then blended to form the final prediction. One model contained two parallel branches which each contained one or two RNNs, and the other consisted of only one branch. Each model was fed into a fully-connected neural layer to predict which class the signal belonged to, and the results from each were blended with a multi-level perceptron. The idea behind this was to boost classification accuracy by combining the predictions made by both models [29]. The results yielded F1-scores up to 97.1% when used with data from a number of different databases.

Hou et al. [39] used an SVM to classify their data, but they used an LSTM-based auto encoder as a feature extraction tool. The model was designed to find a set of features which minimised the reconstruction error (when decoding the features back to an ECG waveform) using the LSTM network.

Finally, Yildrim [40] used a bidirectional LSTM model in their analysis. In this, the data was analysed in both a forwards and backwards directions, as this can give better results in some problem areas. This was compared to a similar model with a unidirectional LSTM, and the accuracy in both cases was found to be above 99%.

2.2.3.4. Other Classification Methods

KNN algorithms can also be used to classify ECG recordings. Bouaziz et al. [41] used a KNN algorithm to classify 5 classes of ECG signal with a 98.71% accuracy. By using a fuzzy KNN, Castillo et al. [42] was able to obtain results of 95.33% for 5 signal classes, and raised this to 98% by combining the output of the KNN classifier with two other ANN classifiers in a 'fuzzy interference system'. Zhao et al. [24] achieved a 95% accuracy with a KNN network, after subjecting the signals to a robust pre-processing method (involving noise elimination, heartbeat normalisation and quality measurement).

Castillo et al. [24] investigated the effect of the integer k on the effectiveness of the classifier. Larger k values reduced the effect of noise, however they also blurred the boundaries between classes [24]. For this reason, Castillo et al. [24] found k = 1 to be the optimal solution, even though k = 1 and k = 3 produced similar results. Other studies found k = 3 was the best choice for their methods [41,42]. Furthermore, selecting k as an odd number was advised, since it prevents the issue of tied votes [41].

Random Forest algorithms (RaF) are comprised of recursively built classification trees, each of which casts a unit vote to determine the classification of input data [34]. The classification is the class which wins the most votes out of the entire forest. RaFs are resistant to noise, and not subject to overfitting, giving them good performance on a number of practical problems [34].

However, the reported results from RaF classification are mixed. Some studies found good results [34,43,44], and others found poorer results [36]. Emanet [34] claimed RaF to be fast, have excellent performance and no cross validation, making them useful for long-term ECG beat classification. Conversely, Zhang et al. [36] noted that RaF generalise poorly, making them far less effective than other methods, such as an SVM.

Yet more methods are mentioned in the literature. Hu et al. [6] used a decision tree algorithm to classify between AF and normal ECGs with high success. Celin and Vasanth [7] mentioned the use of an Adaptive Boosting algorithm, ANN and a Naïve Bayes classifier, and Jambukia et al. [4] briefly reviewed a number of neural networks.

2.2.4. Comparison of Methodologies

Table 1 summarises the results published in the literature for a range of classification methods. This table has been sorted by year, simply to make it easier to display cases where one study tested multiple methods.

Table 1: Comparison of Results in the Literature

	Classification Methodology			Reported Performance	
Researcher	**Number of Features**	**Processing and Feature Extraction**	**Classifier Method**	**Performance Measures**	**Average Performance**
M. Rashed-Al-Mahfuz et al. (2021) [30]	5	CWT and RR intervals	CNN	Accuracy SE SP AUC	99.90% 99.90% 99.98% 99.94%
	2 3 4 5	HHT and RR intervals	CNN	Accuracy Accuracy Accuracy Accuracy	95.75% 89% 81.38% 72.70%
T. Wang et al. (2021) [26]	5	CWT	CNN	Accuracy PPV SE F1-Score	98.74% 70.75% 67.47% 68.76%
Z. Dokur and T. Olmez (2020) [37]	11	--	CNN trained with Walsh Functions	Success	80%-100%
B. Hou et al. (2020) [39]	5 (heartbeat subtypes)	LSTM-based auto encoder	SVM	Accuracy Sensitivity Specificity	99.74% 99.35% 99.84%
	5 (heartbeat classes)	LSTM-based auto encoder	SVM	Accuracy Sensitivity Specificity	99.45% 98.63% 99.66%
Y. Hu et al. (2020) [6]	2	signal splitting, bandpass filtering	Decision Tree	Accuracy Sensitivity Specificity	98.9% 97.93% 99.63%
H. Li et al. (2020) [35]	6	Wavelet packet decomposition	PSO-SVM	Accuracy SE SP PPV	97.78% 97.78% 99.63% 97.87%
			GA-SVM	Accuracy SE SP PPV	98.33% 98.33% 99.72% 98.42%
			GS-SVM	Accuracy SE	98.89% 98.89%

				SP PPV	99.81% 98.92%
O.S. Lih et al. (2020) [38]	4	--	CNN-LSTM	Accuracy SE SP PPV	98.51% 99.30% 97.89% 97.33%
S. Saadatnejad et al. (2020) [29]	5	RR interval, WT	RNN with LSTM	F1-score (dataset A)	97.1%
				F1-score (dataset B)	96.8%
				F1-score (dataset C)	95.5%
J. Huang et al. (2019) [18]	5	Spectrogram	2D-CNN	Accuracy	99.00%
	5	Spectrogram	1D-CNN	Accuracy	90.93%
Y. Zhang et al. (2019) [36]	2	Various feature extraction methods	KSVM	Accuracy	89-92%
			LS-SVM	Accuracy	91-92%
			RaF	Accuracy	89-91%
F. Bouaziz et al. (2018) [41]	5	Wavelet denoising, DWT	KNN	Accuracy	98.71%
S. Celin and K. Vasanth (2018) [7]	2	Bandpass filtering	SVM	Accuracy SE SP	87.5% 75% 100%
	2	Bandpass filtering	Adaptive Booster	Accuracy	93%
	2	Bandpass filtering	ANN	Accuracy	94%
	2	Bandpass filtering	Naïve Bayes	Accuracy	99.7%
C. Venkatesan et al. (2018) [17]	2	Delayed error normalised LMS, and DWT	SVM	Accuracy	96%
O. Yildrim (2018) [40]	5	None	Unidirectional LSTM	Accuracy	99.25%
	5	None	Bidirectional LSTM	Accuracy	99.39%
M. Kropf et al. (2017) [44]	4	QRS detection, and other time-domain features	RaF	F1-score	81%
R. Mahajan et al. (2017) [43]	4	Genetic Algorithm	RaF	Accuracy	82.7%
Z. Zhao et al. (2013) [24]	--	highpass filter, wavelet	KNN	Accuracy	95%

		thresholding, heartbeat normalisation, quality measure			
O. Castillo et al. (2012) [42]	5	Bandpass filter, segmentation, heartbeat normalisation	Fuzzy KNN	Classification rate	95.33%
N. Emanet (2009) [34]	5	DWT	RaF	Accuracy	99.8%
MathWorks [25]	3	Wavelet decomposition	SVM	Accuracy	97.96%

2.3. Review Conclusions

A number of conclusions can be drawn from the literatures review. These suggest suitable means of pre-processing, feature extraction and classification of ECG signals.

First, in terms of pre-processing, noise removal is a necessary first step. Bandpass filtering is easy to implement, so may be a good starting point, although wavelet denoising has shown promising results. These processes should be tested, with further research undertaken as required.

The length of signal segments to be analysed does not seem to play a major role, since different methods have reported success with >30 seconds, 10 seconds and even shorter segments. Since some arrhythmias, like AF, are easily distinguished with variations in RR interval distances, it may be best to use longer signals to capture more of this information. Conversely, shorter signals reduce computational power and time, and may be preferable for this reason.

Second, feature extraction can be completed in a number of different ways, for both features in the time and the frequency domains. For some arrhythmias, it may be sufficient to extract only time- or frequency-domain features, but with the growing popularity of image-based classification methods (like the CNN), it is worth considering the time-frequency information available in spectrograms. A number of the other techniques identified may be worth comparing as well.

Finally, ECG classification can utilise a broad range of techniques, and a wide range of modifications to each of these techniques. SVMs have been identified as a suitable starting point, and should be relatively straight-forward to implement. However, CNNs look promising in the literature. These classifiers also have the advantage of being able to classify signals from the spectrogram images, which are easy to obtain with MATLAB. Alternatively, LSTM networks are well-suited to time-series analysis, and can provide the benefits of a neural network with lower processing requirements than CNNs. These methods should be trialled and compared for effectiveness. If further classification methods are required, more research should be completed to identify appropriate options.

It is worth mentioning that although most papers quoted the accuracy of their classification method, it is rarely suitable to summarise the results with this single metric. The precision and recall (or similarly sensitivity and specificity) give more insight into the actual effectiveness of a classifier, and the F1-score combines these and may be a more effective single measure [29]. For example, a classifier which classifies all signals as normal will have a good recall for this, but a poor precision since it also classified many signals incorrectly. Suppose this classifier was used with a data set of 80% normal signals and 20% abnormal. Here, classifying all signals as normal would give a high accuracy, even though common sense tells us this classifier is no use at all. Hence papers which only quote an accuracy value may not be providing enough information about their results. For example, consider the results by Wang et al. [26] in Table 1, in which the accuracy is an impressive 98.74%, but the F1-score is only 68.76%, demonstrating the model is not as effective as the accuracy suggests. Ensuring the data set is roughly even, such as by copying the signals in both the test and training sets, may help to mitigate this.

Note also that the comparisons made in Table 1 do not take into account all information possible. For example, computational power and time have not been considered here, although they are very real constraints for this project. As these are not limiting factors in this project (as Adapt can be used to run intensive code overnight, for example), further research into this may be completed if absolutely necessary.

In summary, based on the findings of the literature review, the project should progress in the following manner:

1. Begin by applying the raw PhysioNet data [2] to an existing processor which makes use of an SVM [25];
2. Pre-process the data using existing or simple pre-processing techniques (i.e., band-pass filtering), and apply this data to the existing processor;
3. Experiment using other techniques to classify the data, such as an LSTM network and a CNN;
4. Compare different pre-processing and feature extraction techniques using the various classifiers; and,

5. Compare the results of each methodology, and develop a “best” combination of pre-processing, feature extraction and classification.

The effectiveness of the “best” pre-processing and classifier combination will be compared to those quoted in the literature, to assist in determining the final conclusions of the project.

3. Method

The project has taken the following form:

1. Gained a basic understanding of heart disease and ECGs;
2. Performed preliminary research on ECG analysis using ML techniques;
3. Wrote a one-page review of the topic to consolidate understanding;
4. Identified a database and an existing SVM classifier, and examined the effectiveness of the classifier without pre-processing;
5. Identified other processing steps, including spectrograms and the MoV, and gained an understanding of how these could help with the project;
6. Explored other classification techniques, including a CNN and LSTM model; and,
7. Tested the three models with different pre-processing steps, and compared the results to identify the method with the highest performance.

The following sub-sections will discuss how each of these steps was carried out, and any major challenges involved. See Section 4: Results for the results produced, and Section 5: Discussion for a discussion on the results and method.

Production of deliverables, such as the interim thesis and seminar, and the project wiki page and Ingenuity exhibit will not be discussed.

3.1. Heart Disease and ECG Understanding

A basic understanding of the human body, and in particular the cardiovascular system, was previously obtained during courses undertaken for Biomedical Engineering. However, heart disease had not yet been analysed. It was important to gain a basic understanding in this area to give insight into the variety of problems which could be identified in the ECG recordings.

At this time, PhysioNet [2] was identified as a suitable database from which to obtain ECG recordings. The database contained 8528 ECG recordings classified into four groups: normal rhythm, atrial fibrillation (AF), other rhythm, and noisy signal. Hence, both normal rhythm and AF were researched further.. It was decided that only this database would be used throughout the project. This would allow meaningful comparisons to be made between different methodologies.

3.2. One-Page Review

A one-page review was developed to succinctly consolidate and demonstrate understanding of the topics covered at that time. Although lacking much technical detail, this review identified a number of good references and acted as a starting point for the literature review.

A copy of the one-page review is included as-is in Appendix A.

3.3. Existing SVM Classifier in MATLAB

An example MATLAB script which used both wavelet de-noising and an SVM classifier was identified [25]. This code was run without change to replicate the results provided by the example. This was deemed to be a suitable benchmark to compare future results to.

The data collected from the PhysioNet database [2] was then processed to be classified by this algorithm. Processing involved ensuring all signals were of the same length and required data types. The example code also had to be modified extensively to be compatible with our data. These required changes were:

- Changing the classes to match those in the data used;
- Adding additional variables to compensate for the change in signal classes; and,
- Modification of window length as data contained fewer data points.

Further discussion on how these changes were made is included below, and the results produced are included in the Results section. A copy of the original code can be found here: https://au.mathworks.com/help/wavelet/ug/ecg-classification-using-wavelet-features.html [25].

3.3.1. Preparing the Data

The data downloaded from the PhysioNet database [2] comprised of a folder of individual MATLAB vectors. Each vector represented one ECG recording. These recordings had a consistent sampling frequency (300 Hz [2]), but did not have a consistent length. As such, the first step in preparing the data was to select a data length. A number of options were considered:

1. Truncating each ECG recording to the length of the shortest recording (2712 samples $\approx$ 9 seconds, in this case);
2. Selecting a length and truncating all recordings longer than this while removing all recordings shorter than this:
   i. Length of 3000 samples, equivalent to 10 seconds;

ii. Length of 6000 samples, equivalent to 20 seconds;

iii. Length of 9000 samples, equivalent to 30 seconds; and,

3. Duplicating the recording as many times as required to ensure each signal had the same number of points as those in the example code (i.e. 65536 data points).

Other studies had used 10 seconds of ECG recording (or split recordings into 10 second blocks) [20,22,23,24], so using 3000 samples was chosen for this reason. Over half of the signals had 9000 data points, so both 6000 and 9000 samples could be used without significantly reducing the size of the data set.

The PhysioNet data [2] contained four different ECG classifications. These were normal rhythm (N), atrial fibrillation (A), other arrhythmia (O), and noisy recording (~). After some experimentation, the number of different classes was also altered to investigate how this would affect the results:

4. All four data classes were included;

5. The noisy data class was removed, the other three were included; and,

6. The noisy data was removed and the AF and other arrythmia classes were combined into a single 'abnormal' class.

This would allow investigation as to whether the number of classes the SVM had to discriminate between had an impact on the results it was able to produce.

It was also discovered that the way the example code separates the data into a training set and a test set requires that the data be sorted in order of its class (i.e. all normal signals first, then all AF signals, etc.). Hence, after the processing steps mentioned above, the data was sorted according to its classification.

3.3.2. Modifying the Example Classes

The example code contained ECG recordings of three different types: normal sinus rhythm (NSR), arrhythmia (ARR) and congestive heart failure (CHF), whereas the data downloaded from PhysioNet [2] contained four different types. Due to this, there were a number of cases where the example code classes had to be changed from {ARR, CHF, NSR} to {A, O, N, ~}.

Other points in the code required additional variables to be added to enable the fourth class to also be identified. This also required altering matrix indices which used to go up to 3, but now needed to include 4.

3.3.3. Magic Numbers

The example code contained a number of parameters which were very specifically chosen to fit with the data, yet were hard-coded. The most obvious of these (since it produced an error) was the window length used for feature extraction. This was hard-coded as 8192 samples, or one eighth the length of the supplied signals. This was modified to a formula which found the (rounded-down) length of an eighth of the length of an ECG data signal being processed.

Other magic numbers were included in the code, such as the autoregressive model order, and the polynomial order of the SVM classifier. These numbers do not throw an error when the code is executed, however they may not be the optimal parameters for the data used.

3.4. Pre-Processing

A number of pre-processing techniques were examined. These included the Pan-Tompkins algorithm, wavelet denoising and the MoV. Each of these were applied to the data, and tested on the classifiers.

3.4.1. Pan-Tompkins Algorithm

One method of pre-processing which was identified was the Pan-Tompkins algorithm [4,22]. This involves the following steps:

1. Removal of DC offset;
2. Band pass filtering to reduce noise from the ECG signal;
3. Differentiation to find high slopes which usually identify R-peaks and suppresses low frequency components of the P and T waves;
4. Squaring to further enhance high frequency components;
5. Averaging the signal with an averaging function; and,
6. Moving window integration to extract the slope of the R wave.

Each of these steps were completed using MATLAB. At each step the results produced were compared to those in [22] to verify the correct outcome was achieved. This process is described in more detail below.

The first step was to remove the DC offset of the ECG by subtracting the mean from the signal. The next step was to apply a bandpass filter to the data to remove noisy high-frequency components, and low frequency artifacts (such as breathing) from the ECG recordings. In theory, this would make it easier to classify the ECG signals.

The signal then needed to be differentiated to find the high slopes which are indicative of the R-peak on the ECG waveform. This was done in MATLAB using the gradient function. This was then squared to make these high frequency changes even more noticeable, and ensure the whole signal was positive.

The averaging function was realised using a moving window filter. A window size of 15 was found to be sufficient. This function smoothed the spikes in the signal into gentle peaks. The signal integration was also completed with a moving window. In this case a window of width 30 was moved across the signal. The result was further smoothing of the signal peaks.

The results of this process were then able to be analysed.

3.4.2. Scalograms

Scalograms were also identified as a possible tool to help with the classification of ECG signals. Research was done to identify the information which could be extracted from the spectrogram, and how to apply these to an ECG signal. The CWT function in MATLAB was used to plot the scalograms of the ECG signals. Scalograms are powerful tools when classifying images, hence they were later used with the CNN classifier.

3.4.3. Moment of Velocity

The MoV was also identified as a possible tool to make the data more suitable to feature extraction. The source code for calculating the MoV of an ECG was located [45], and this was modified to fit the data. This function was used on each signal in turn to produce the MoV of the ECG recording. The data was then ready to have features extracted and be classified.

3.5. Neural Network Classification

Along with the SVM, two types of neural networks were examined. These were a CNN which used spectrogram images to classify ECG signals, and an LSTM network which used time-series data.

An LSTM network designed to classify ECG signals was identified [14]. This was also created by MathWorks, so the code was well-suited to quick modification. Parameters were altered as need be (this was quite minimal), and the network was used to classify the data. The LSTM network used contained a bidirectional LSTM layer, meaning the signal was analysed from both directions. First, the raw data was used to train and classify the network, then feature extraction was used to improve the results. The features extracted were the instantaneous frequency and entropy of the data. The network was tested with the raw ECG data, denoised ECG data, and the MoV of the data, and the results of each were recorded for comparison. The LSTM network was also tested using the Pan-Tompkins peak detection, although no other classifiers were.

Meanwhile, Hien Long completed a similar process with the CNN. This classifier made use of a pre-trained SqueezeNet network. Transfer learning was used to adapt the classifier to ECG data. Hien Long also made further modifications to the SVM classifier to achieve better results.

3.6. Comparison of Models

Finally, the classification models were compared with one another. To do this, a single measure was chosen, this being the F1-score [29] of the AF class. This would allow meaningful comparisons between each methods' ability to detect AF to be carried out. The F1-score can be calculated by the following formula:

$$F1score = 2 \times \frac{precision \times recall}{precision + recall}$$

Where, given TP (true positive), FP (false positive) and FN (false negative) values:

$$precision = \frac{TP}{TP + FP} \qquad \text{and,} \qquad recall = \frac{TP}{TP + FN}$$

Alternatively, the precision and recall values are the values in the circled squares of the confusion matrix in Figure 10 below.

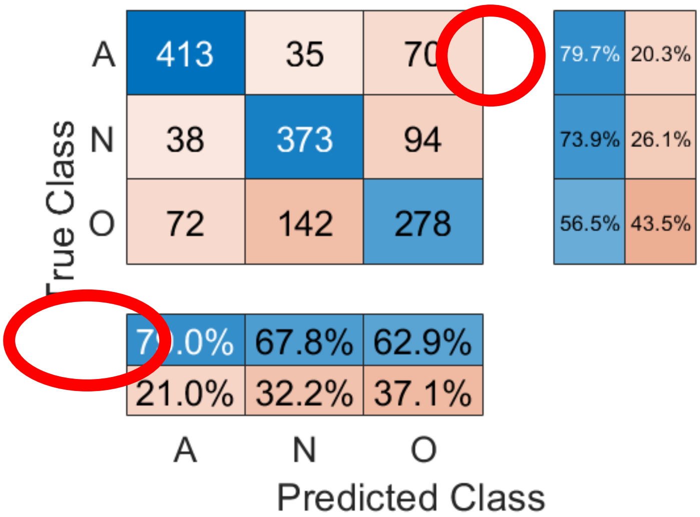


Figure 10: Calculating F1-score from confusion matrix.

4. Results

4.1. MATLAB SVM Classifier Example

Without altering any parameters or data, the identified MATLAB classifier example produced a test accuracy of 97.96% [25]. This classifier was given 162 ECG signals, of which 70% were used to train the machine, and the other 30% were used for testing. The example contained three different types of signals: NSR, ARR and CHF. In total, 96 of these recordings were from patients with ARR, 30 with CHF, and 36 recordings with NSR.

The features of each signal were extracted using wavelets. This decreased each signal from 65536 data-points in the time-domain to just 190 features [25]. This made the data quicker and easier to process. There are differences in each feature between the classes. For example, Figure 11 shows the variance in the second-lowest frequency wavelet sub-band [25]. While no one feature alone is enough to separate all classes, the idea is to have a rich enough set of features to make this process accurate.

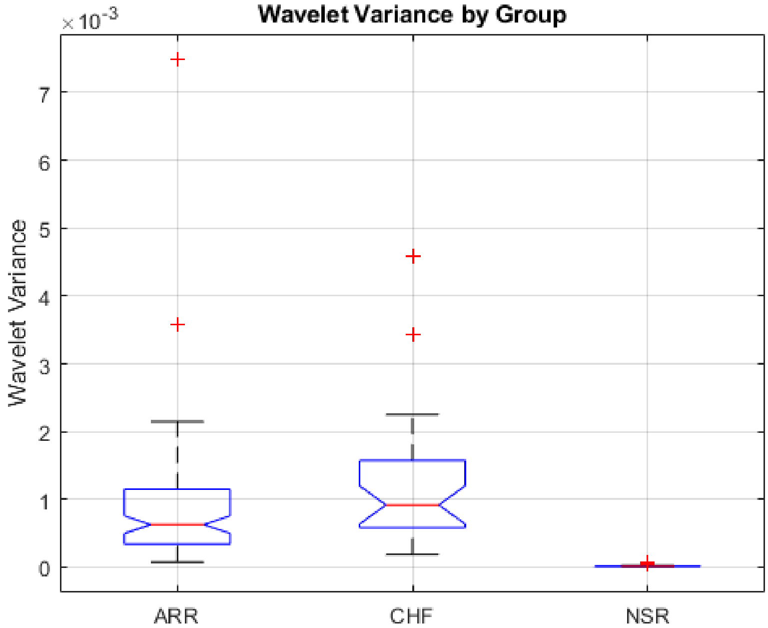


Figure 11: Wavelet Variance by Group in MATLAB Example [25]

When the example code was run exactly as-is, the results in Table 2 were produced. These results show how well the trained classifier sorted the testing set of data. All ARR signals were correctly identified, and no other signals were accidentally assigned this label, although one CHF signal was misclassified as an NSR signal.

These results were considered a good baseline for further study.

Table 2: MATLAB Example Classifier Results [25]

```
         Precision    Recall    F1_Score
         _________    ______    ________

ARR         100         100        100
CHF         100       88.889     94.118
NSR       91.667        100      95.652
```

4.2. SVM Classifier Results

After verifying the example code, the next stage of analysis was to fit the results collected from the PhysioNet database [2] to the classifier. In essence, this required selecting a time interval (or equivalently number of data points) and truncating or extending the data to fit this, deciding how many classes of signal to classify for, and then altering the example code to fit the data prepared.

As outlined in the Method, a number of different cases have been tried. Table 3 summarises these results, and the precision/recall/f1-score table for each type tested can be found in Appendix B.

Note that execution time and CPU and memory requirements were not recorded as all cases were able to run successfully on the PC (even while other processes were running, such as a web browser and text document). However, it should be noted that data sets with more samples took longer to process than ones with less samples, and that memory requirements were also higher for the longer signals.

Table 3: Summary of Results from the Example MATLAB SVM Classifier

Name	Samples per Signal	Number of Classes	Number of Signals					Accuracy
			Total	A	O	N	~	
MATLAB Example Data	65536	3	162	-	-	-	-	97.96%
Truncated to Shortest ECG Recording	2712	4	8527	738	2456	5049	284	62.98%
Truncated to 3000 Samples	3000	4	8511	737	2452	5039	283	62.10%
Truncated to 3000 Samples	3000	3	8220	737	2452	5031	0	64.20%
Truncated to 6000 Samples	6000	4	7911	654	2348	4741	168	66.02%
6000 samples, Three Classes	6000	3	7743	654	2348	4741	0	66.88%

6000 samples, Two Classes	6000	2	7743	0	3002	4741	0	70.34%
Truncated to 9000 Samples	9000	3	7415	625	2262	4528	0	65.87%
9000 samples, All Classes	9000	4	7560	625	2262	4528	145	65.06%
Extended to 65536 Samples	65536	4	8527	738	2456	5049	284	63.78%

4.3. Modified Pan-Tompkins Algorithm

The results of using the modified Pan-Tompkins Algorithm to process one signal is shown in Figure 12.

Panel (a) shows the original ECG signal with the DC component removed. This means the baseline of the signal rests at 0.

Panel (b) shows the ECG after filtering with a bandpass filter. This has minimal effect on the appearance of the signal since there was little noise in the original signal.

Panel (c) shows the signal after derivative filtering. This makes the large peaks (R-peaks of the ECG) stand out, and hides the smaller peaks due to the P and T waves.

Panel (d) shows the signal after squaring. This further highlights the large peaks of the R waves.

Panel (e) shows the signal after averaging. This converts the signal into a series of peaks corresponding to the R-peaks.

Panel (f) shows the signal after integration which smooths these peaks.

Figure 13 then shows how the signal produced at the end of this sequence corresponds to the recorded ECG signal. Notice that each of the peaks corresponds to a QRS complex in the signal and that other peaks (even the noise after the first beat) are not counted. Hence, this algorithm is able to identify the R-peaks in an ECG recording. This can be a useful first step in classifying whether or not a signal is abnormal since irregular heartbeats often signify an abnormal condition.

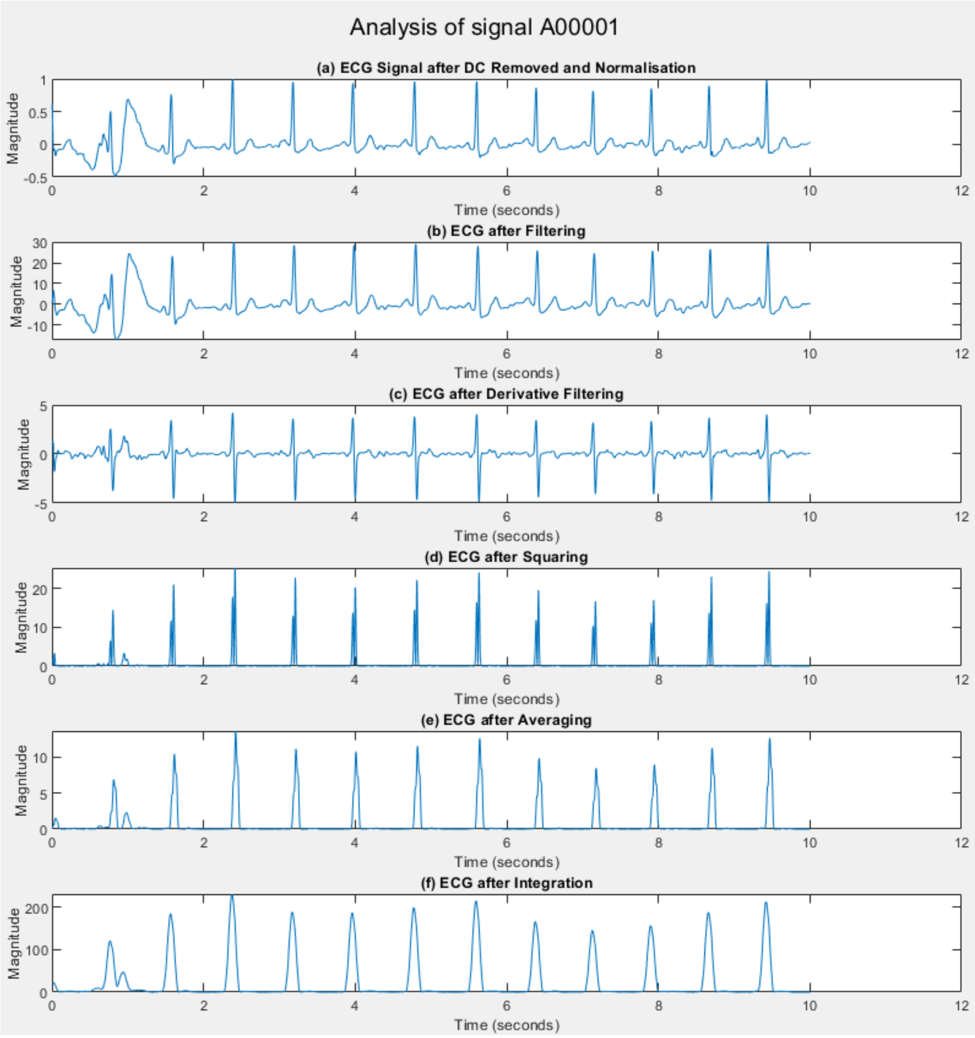


Figure 12: Modified Pan-Tompkins Algorithm Stages

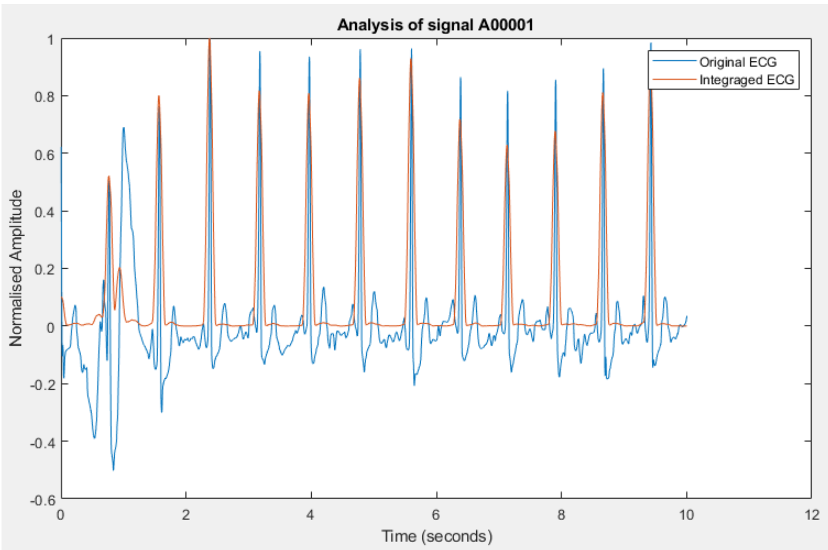


Figure 13: Identified R-Peaks of ECG Signal

4.4. Improved SVM Classifier Results

The SVM classifier was tested on the raw ECG data, wavelet denoised data, and the MoV. For each case, two sets of features were examined. Figure 14 shows the confusion matrices of the results produced when only a set of 12 time-domain features were used for classification. Conversely, Figure 15 shows the confusion matrices when 169 features from the time- and frequencies-domains, plus statistical features, were used. The F1-scores for each case are shown in Table 4.

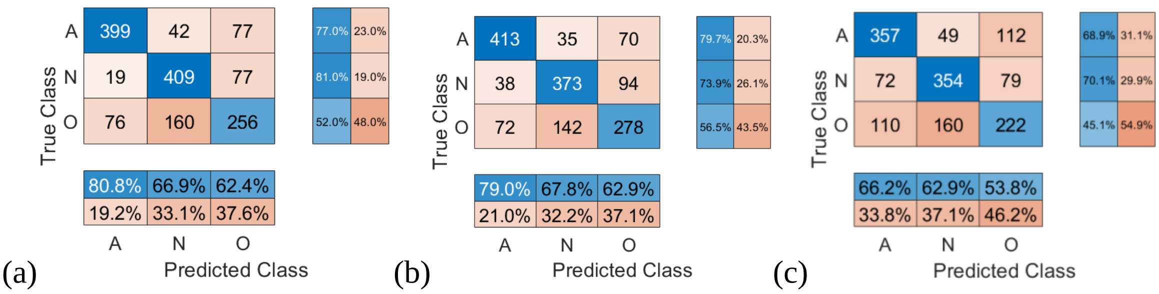


Figure 14: Confusion matrices for SVM classification with only time-series features, (a) raw data, (b) wavelet denoised, and (c) MoV.

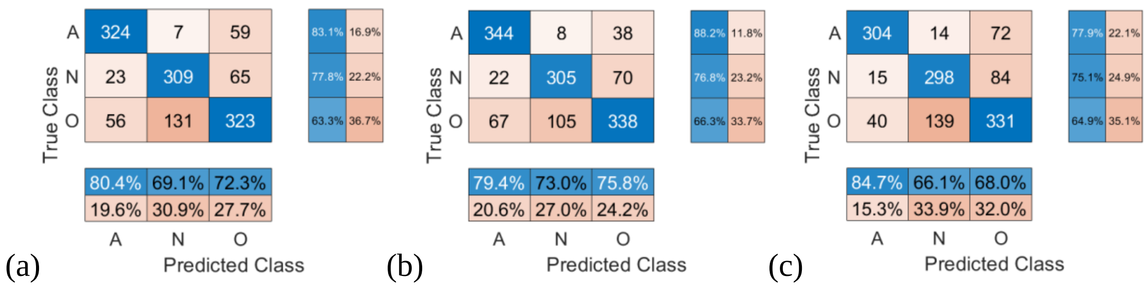


Figure 15: Confusion matrices for SVM classification with all 169 features, (a) raw data, (b) wavelet denoised, and (c) MoV.

Table 4: F1-scores for the SVM with different feature sets.

Pre-processing	12 Time-domain Features	169 Multi-domain Features
Raw data	0.785	0.814
Wavelet denoising	0.794	0.836
Moment of velocity	0.675	0.760

4.5. LSTM Classifier Results

The LSTM classifier was tested on the raw ECG data, denoised data, and the MoV of the data. As a point of comparison, the LSTM was trained and then used to classify ECGs based only on their time-series recording (i.e. no feature extraction). The confusion matrix is shown in Figure 16(a), and as can be seen, the results are poor, demonstrating the importance of appropriate feature extraction.

Figure 16(b) shows the confusion matrix for the raw data with feature extraction, and Figure 16(c) and Figure 16(d) show the result with wavelet denoising and MoV pre-processing steps, respectively.

The F1-scores were calculated to be 0.507, 0.686, 0.817, 0.657, respectively.

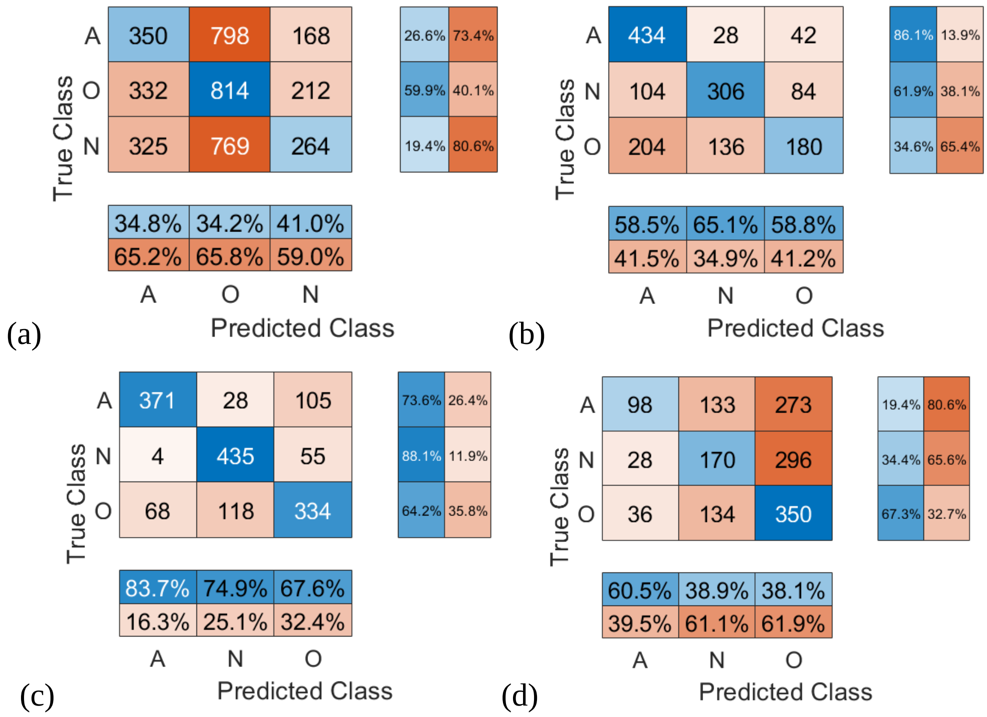


Figure 16: Confusion matrices for LSTM classification, (a) no features, (b) raw data, (c) wavelet denoised, and (d) MoV.

4.6. CNN Classifier Results

The CNN was also tested on the raw ECG data, denoised data and the MoV. The confusion matrix of each is expressed in Figure 17 below. The F1-scores were calculated to be 0.816, 0.835, and 0.771, respectively.

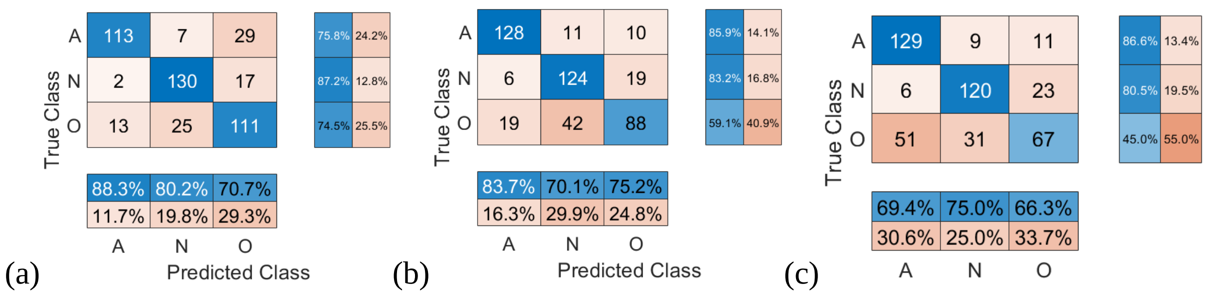


Figure 17: Confusion matrices for CNN, (a) raw data, (b) wavelet denoised, and (c) MoV.

4.7. Comparison of Methods

The results of the three classification methods, tested with three different types of data, were compared to one another to identify a best result. The F1-score for identifying the AF class achieved by each is shown in both Figure 18 and Table 5 below. These results are discussed in Section 5: Discussion.

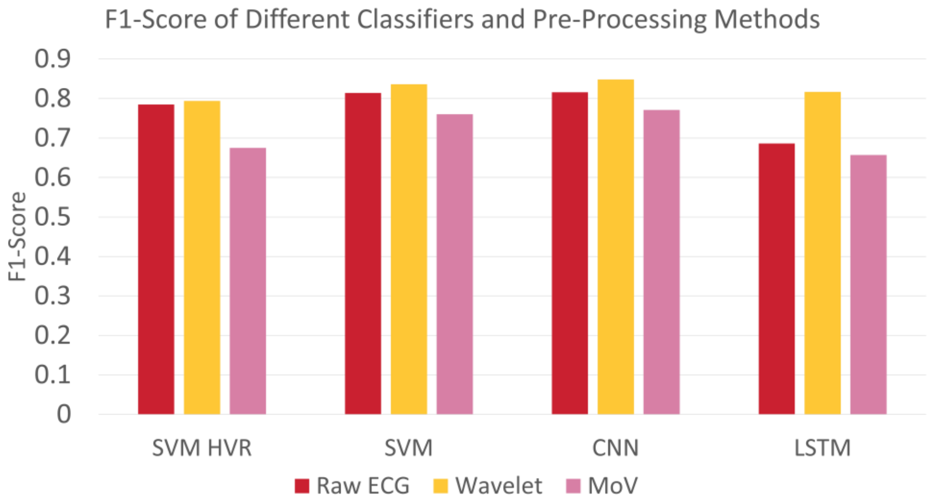


Figure 18: F1-score of classification results.

Table 5: Details and Results of each Classification Method.

Classification Method	Pre-processing Stages	Features Extracted	F1-score
SVM	Raw ECG data	HRV	0.785
SVM	Wavelet Denoising	HRV	0.794
SVM	Wavelet Denoising and Moment of Velocity	HRV	0.675
SVM	Raw ECG data	Time and Frequency Domain, Signal Quality, and Non-linear and Morphological Features	0.814
SVM	Wavelet Denoising	Time and Frequency Domain, Signal Quality, and Non-linear and Morphological Features	0.836
SVM	Wavelet Denoising and Moment of Velocity	Time and Frequency Domain, Signal Quality, and Non-linear and Morphological Features	0.760
CNN	Raw ECG data	Spectrogram	0.816
CNN	Wavelet Denoising	Spectrogram	0.848
CNN	Wavelet Denoising and Moment of Velocity	Spectrogram	0.771
LSTM	Raw ECG Data	None - computed on raw ECG data	0.507
LSTM	Raw ECG data	Instantaneous frequency, Entropy	0.686
LSTM	Wavelet Denoising	Instantaneous frequency, Entropy	0.817
LSTM	Wavelet Denoising and Moment of Velocity	Instantaneous frequency, Entropy	0.

5. Discussion

This section discusses processing considerations, the effect of different pre-processing methods, and the performance of the different classifiers. The overall results and the implications of these are then discussed.

5.1. Considerations

A number of parameters can be altered to optimise the data, or the results possible from the data. This optimisation could be in terms of accuracy, or computational considerations.

One such parameter was the length of the data used. The initial trials on the MathWorks SVM [25] showed little difference between the length of the signals and the accuracy of the model, as demonstrated by Figure 19 (varies less than 4%). Hence, it was decided that 10-second segments would be used, which corresponds to 3000 samples in the data. This shorter length was chosen as it could reduce computational power and time, and was supported by a number of sources [20,22,23,24].

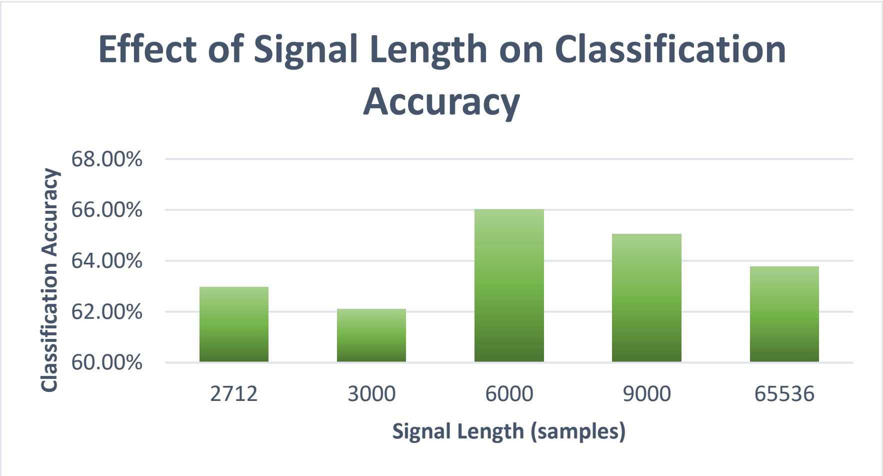


Figure 19: Effect of signal length on classification accuracy of SVM classifier (4 classes).

Another consideration was the number of classes to use. The database contained 4 classes, being Normal (N), Atrial Fibrillation (A), Other abnormal (O) and noisy (~). As shown in Figure 20, the results demonstrated that regardless of the signal length, including all four classes produced a poorer result than using three or two. It was decided that as the noisy condition does not provide any information on the heart condition of the patient, these signals would be omitted from the analysis.

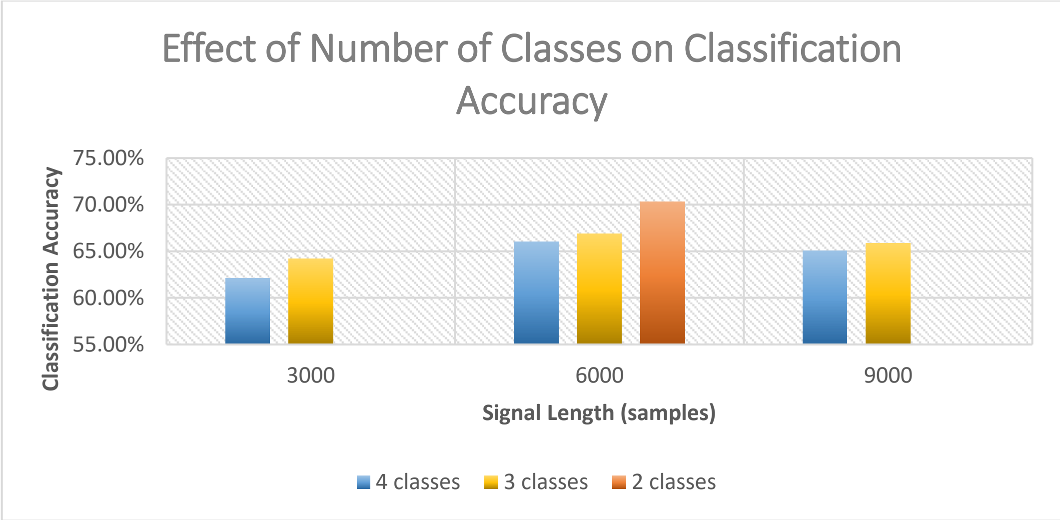


Figure 20: Effect of number of classes on classification accuracy of SVM classifier

It was also decided that a specific abnormal condition should be identified from the others. Hence, all results are reported in relation to the AF result, as this condition was already labelled in the data. In this way, the results show the ability of the classifier to identify AF from a set containing both normal signals and ECGs with other abnormalities.

Finally, it was important to consider how long the model was trained. If the model is trained for a short period of time, it may not be trained well enough, and produce poor results. Conversely, if a model is trained too long, it can learn to overfit the data, and also produce poor results.

5.2. Pre-processing

A number of pre-processing methods were analysed. Each of these transform the ECG signals in some way. Here, wavelet denoising, Pan-Tompkins algorithm, and the MoV were considered, as well as using the un-processed raw data. A comparison of each for a normal ECG waveform is shown in Figure 21. The Pan-Tompkins algorithm is effective at finding the QRS complex of a given ECG signal (see Figure 13 for overlaid demonstration).

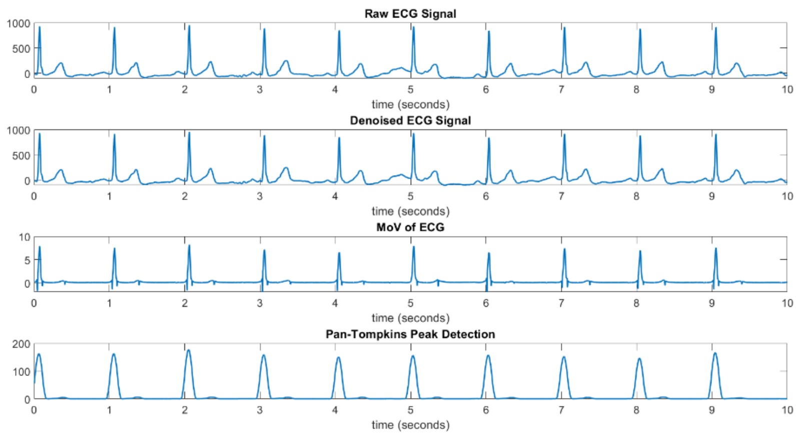


Figure 21: Comparison of (a) raw ECG, (b) wavelet denoised, (c) Pan-Tompkins, and (d) MoV.

A comparison on how these impacted the classification of the signals when using the LSTM network is shown in

Figure 22. The wavelet denoising produced a notable better result than all the other options, with the Pan-Tompkins QRS detection and the MoV both performing poorly. This can be explained by the shape of the waveform produced. In both of these cases, the location of the QRS complex is highlighted, but other information such as that of the P-waves and T-waves, is removed. Especially in the case of identifying AF, this is not helpful, as the shape of the P-wave and other small waves is critical in the diagnosis of AF. Hence, it is expected that these two techniques would not perform well under these conditions. The wavelet denoising does however produce a better result, as noise is removed from the signal, without disrupting the finer features on the signal.

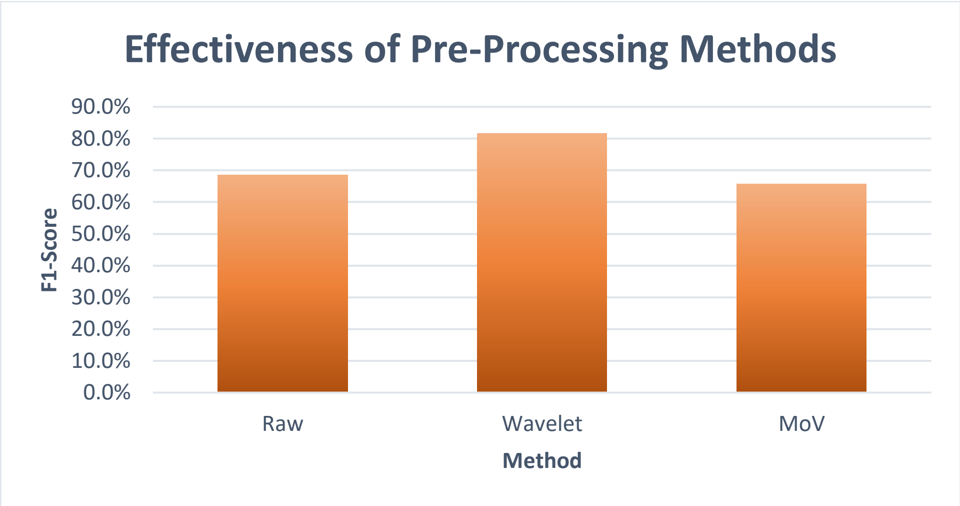


Figure 22: Effectiveness of pre-processing techniques on LSTM classification.

Notice for the other classification methods, and in particular the CNN, the wavelet denoising has a minimal impact on the results produced (compared with the raw signal).

5.3. Feature Extraction and Classification Methods

Three classifiers were compared during the completion of this project, an SVM, a CNN and an LSTM network. The primary discussion here relates to the performance of the methods, although a quick comment on the computational time is mentioned.

The F1-score of each of the classifiers is shown in Figure 18 (Section 4.7). The results show the CNN was the most effective at identifying AF, with the SVM results following closely. The LSTM did not perform as highly.

Two different cases of the SVM were examined. The first (referred to as SVM HRV in the Figure 18) was used with only 12 time-domain features, whereas the second (referred to as SVM), used 169 features from the time- and frequency-domains and other statistical information. These two results are very similar, and the numerical results shown in Table 4highlight this. This demonstrates that in distinguishing AF, time-domain features provide more information than frequency-domain and other statistical features. Bear in mind, this result is for AF, and may not be the case for other forms of CVD.

The LSTM produced significantly lower results than the other two classifiers. This is likely due to the features which the model was designed to extract. In the classifier used, the instantaneous frequency and entropy were extracted, and reduced to 63 samples each. Not only were these not the most useful of measures, but they held very similar information. This also meant that the results were very similar when both features were used, and when only one of the features was used, as demonstrated by Figure 23.

The Receiver Operating Characteristics (ROC) for each model can be seen in Figure 24. The ROC curve is a measure of the effectiveness of the model. The more a curve hugs the left and top edges of the diagram, the better it is able to classify the signals. This shows that the SVM and CNN are both highly effective models.

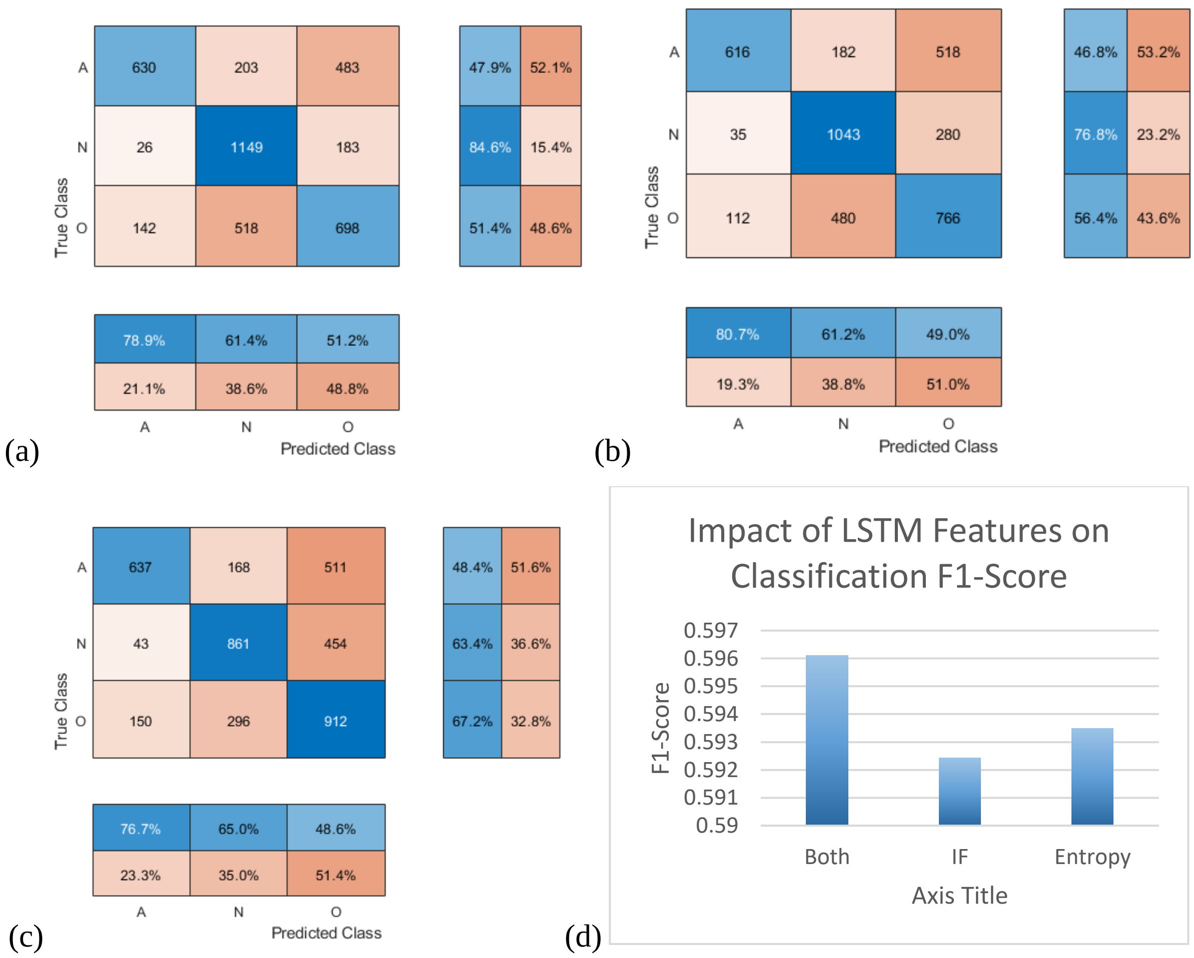


Figure 23: Confusion matrices for the LSTM with (a) both features, (b) just instantaneous frequency, and (c) just entropy, and a comparison of F1-scores (d).

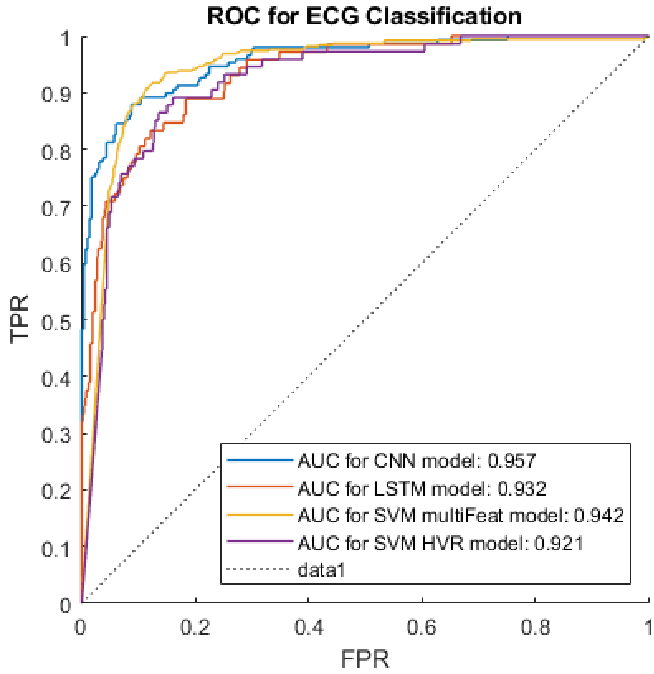


Figure 24: ROC for each model.

5.4. Overall Results

The overall results shown above demonstrate that a ML classifier can distinguish between heart conditions reasonably well. In the results here, only three conditions were considered, but this could be extended to a wider range of abnormalities.

The results achieved by these methods are comparable to some in the literature. Although these results are less effective than some [6,35,38,39], they are comparable to others [7,44], and better than some [26]. Papers which only quoted the accuracy of their method are deemed not suitable for comparison, as accuracy is a misleading metric.

In terms of the project aims, multiple ML methods were used to classify ECG recordings. The most effective was found to be a CNN after data had been subject to wavelet denoising.

6. Possible Future Work

This project could be extended in future in a number of different ways. Possibilities include:

1. Modify the combination of pre-processing, feature extraction and classification techniques;
2. Experiment with further machine learning techniques;
3. Extend the classification to distinguish between a larger number of cardiovascular conditions; or,
4. Evaluate the possibility of including ML diagnosis software such as this in a wearable device.

The results produced here, although quite good, hold plenty of room for improvement. Hence, it would be valuable to extend this project in future, to find other methods which are more effective at classifying ECGs using ML techniques.

7. Conclusions

So, can we teach a machine to be a cardiologist? In terms of teaching a machine to accurately distinguish between different heart conditions, yes this is possible, as evidence by the results presented.

The results show it is highly useful to pre-process the signal in a suitable way, such as by denoising with wavelets, as this improved the results from the raw data case. Other pre-processing methods tried were not effective. The features extracted then need to be suitable to the method used. For example, the time-frequency scalograms were effective when using a CNN classifier, and the time-domain features were useful with the SVM. However, the time-frequency signals extracted by the LSTM were not as useful, and led to poorer results.

The classification methods were compared, and the CNN and SVM were both found to be highly effective at classifying the data. The LSTM was less effective, however this may be due to the features extracted, rather than the classifier itself. The CNN and SVM both achieved an F1-score greater than 80%, meaning the models were both highly effective at identifying AF from a range of normal and abnormal ECGs.

However, if a system like this were to be applied in a hospital setting, for example, further work should be completed to improve the effectiveness of the classifier, and extend it to identify a greater range of cardiovascular abnormalities.

8. Definitions

Term	Meaning
Deep learning	"a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks" from: https://machinelearningmastery.com/what-is-deep-learning/
Machine learning	"the use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyse and draw inferences from patterns in data." from: https://languages.oup.com/google-dictionary-en/
PhysioNet	Database which the examined ECG recordings were downloaded from
P-wave	ECG feature corresponding to contraction of the atria (refer to Figure 1)
QRS Complex	ECG feature corresponding to the contraction of the ventricles (refer to Figure 1)
R-peak	The characteristic, (usually) highest peak on an ECG waveform (refer to Figure 1)
RR interval	The time between subsequent R-peaks in an ECG recording
T-wave	ECG feature corresponding to repolarisation of the ventricles (refer to Figure 1)
VGG16	A convolutional neural network architecture with 16 layers

9. Abbreviations

Abbreviation	Meaning
AF	Atrial Fibrillation
ANN	Artificial Neural Network
ARR	Arrhythmia
AUC	Area Under Region of Convergence Curve
bpm	Beats per minute
CHF	Congestive Heart Failure
CNN	Convolutional Neural Network
CVD	Cardiovascular Disease
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
FIR	Finite Impulse Response
FT	Fourier Transform
GA	Genetic Algorithm
GS	Grid Search Algorithm
HHT	Hilbert-Huang Transform
IF	Instantaneous Frequency
KSVM	Kernel Support Vector Machine
KNN	K-Nearest Neighbour
LMS	Least Mean Squares
LS-SVM	Least Squares Support Vector Machine
LSTM	Long-Short Term Memory
ML	Machine Learning
MoV	Moment of Velocity
NSR	Normal Sinus Rhythm
PPV	Positive Predictive Value
PSO	Particle Swarm Optimisation
RaF	Random Forest
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristics
SC-LNLMS	Self-Correcting Leaky Normalised Least Mean Squares
SE	Sensitivity
SP	Specificity
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
WHO	World Health Organisation

10. References

[1] W.H.O., 2017, *Cardiovascular Diseases (CVDs)*, Accessed: 24 May 2021, [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[2] G. Clifford, C. Liu, B. Moody, L. Lehman, I. Silva, A. Johnson, R. Mark; 2017, *AF Classification from a Short Single Lead ECG Recording – The PhysioNet Computing in Cardiology Challenge 2017*, Accessed: 8 March 2021, [Online], Available: https://www.physionet.org/content/challenge-2017/1.0.0/

[3] P.S. Addison, *Wavelet Transforms and the ECG: a Review*, in Physiological Measurement, vol. 26, 2005; [Online], Available: https://iopscience.iop.org/article/10.1088/0967-3334/26/5/R01/pdf

[4] S.H. Jambukia, V.K. Dabhi, H.B. Prajapati, *Classification of ECG Signals using Machine Learning Techniques: a Survey*, IEEE, 2015; [Online], Available: https://ieeexplore.ieee.org/abstract/document/7164783

[5] ecgwaves.com, *Clinical ECG Interpretation*, 2021, Accessed: 18 May 2021, [Online], Available: https://ecgwaves.com/course/the-ecg-book/

[6] Y. Hu, Y. Zhao, J. Liu, J. Pang, C. Zhang, P. Li, *An Effective Frequency-Domain Feature of Atrial Fibrillation Based on Time-Frequency Analysis*, in BMC Medical Informatics and Decision Making, vol. 20, 2020; [Online], Available: https://link.springer.com/article/10.1186/s12911-020-01337-1

[7] S. Celin, K. Vasanth, *ECG Signal Classification Using Various Machine Learning Techniques*, in Journal of Medical Systems, vol. 42, 2018; [Online], Available: https://link.springer.com/article/10.1007/s10916-018-1083-6

[8] S.R. Messer, J. Agzarian, D. Abbott; 2001; *Optimal Wavelet Denoising for Phonocardiograms*, in Microelectronics Journal, vol. 32, issue 12, 2001, pp 931-941; [Online], Available: https://www.sciencedirect.com/science/article/pii/S0026269201000957

[9] O. Faust, U.R. Acharya, H. Adeli, A. Adeli; 2015, *Wavelet-Based EEG Processing for Computer-Aided Seizure Detection and Epilepsy Diagnosis*, in Seizure, vol. 26, 2015, pp 56-64; [Online], Available: https://www.sciencedirect.com/science/article/pii/S1059131115000138

[10] B. Grossfeld, *Deep Learning Vs Machine Learning: a Simple Way to Learn the Difference*, 7 Apr 2021, Accessed: 24 May 2021, [Online], Available: https://www.zendesk.com/blog/machine-learning-and-deep-learning/

[11] W.S. Noble, *What is a Support Vector Machine?*, in Nature Biotechnology, vol. 24, 2006; [Online], Available: https://www.ifi.uzh.ch/dam/jcr:00000000-7f84-9c3b-ffff-ffffc550ec57/what_is_a_support_vector_machine.pdf

[12] R. Gholami, N. Fakhari, *Support Vector Machine: Principles, Parameters, and Applications*, in Handbook of Neural Computation, 2017, pp 515-535; [Online], Available: https://www.sciencedirect.com/science/article/pii/B9780128113189000272

[13] L. Chang, Z. Zhang, L. Ye, D. Friedrich, *Synergistic Effects of Nanoparticles and Traditional Tribofillers on Sliding Wear of Polymeric Hybrid Composites*, in Tribology of Polymeric Nanocomposites, 2nd ed., 2013, pp 49-89; [Online], Available: https://www.sciencedirect.com/science/article/pii/B9780444594556000039

[14] Mathworks, (2017), *Classify ECG Signals Using Long Short-Term Memory Networks*; [Online], Available: https://au.mathworks.com/help/signal/ug/classify-ecg-signals-using-long-short-term-memory-networks.html

[15] M. Phi; 2018; *Illustrated Guide to LSTM's and GRU's: A step by step explanation*; [Online], Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[16] S. Saha, *A Comprehensive Guide to Convolutional Neural Networks – the ELI5 Way*, 16 Dec 2018, Accessed: 24 May 2021, [Online], Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[17] C. Venkatesan, et al.; *ECG Signal Preprocessing and SVM Classifier-Based Abnormality Detection in Remote Healthcare Applications*; IEEE, 2018; Accessed 20 March 2021; [Online] DOI: 10.1109/ACCESS.2018.2794346

[18] J. Huang, B. Chen, B. Yao, W. He, *ECG Arrhythmia Classification Using STFT-Based Spectrogram and Convolutional Neural Networks*, in IEEE Access, vol. 7, 2019; [Online]. Available: https://ieeexplore.ieee.org/document/8759878

[19] R. He, K. Wang, Q. Li, Y. Yuan, N. Zhao, Y. Liu, H. Zhang; *A Novel Method for the Detection of R-Peaks in ECG Based on K-Nearest Neighbours and Particle Swarm Optimisation*, in EURASIP Journal on Advances in Signal Processing, 2017; [Online]. Available: https://link.springer.com/article/10.1186/s13634-017-0519-3

[20] J. Wang, P. Wang, S. Wang, *Automated Detection of Atrial Fibrillation in ECG Signals Based on Wavelet Packet Transform and Correlation Function of Random Process*, in Biomedical Signal Processing and Control, vol. 55, 2020; [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809419302435

[21] A. Khiter, A.B.H. Adamou Mitiche, L. Mitiche, *Muscle Noise Cancellation from ECG Signal Using Self Correcting Leaky Normalised Least Mean Square Adaptive Filter under Varied Step Size and Leakage Coefficient*, in Traitement du Signal, vol. 37, no. 2, 2020, pp. 263-269; [Online], Available: https://doi.org/10.18280/ts.370212

[22] Y. Palaniappan, V.A. Vishanth, N. Santhosh, R. Karthika, M. Ganesan; 2020, *R-Peak Detection Using Altered Pan-Tompkins Algorithm*, IEEE, 2020; [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9182298

[23] I. Saini, D. Singh, A. Khosla, *Delineation of ECG Wave Components Using K-Nearest Neighbour (KNN) Algorithm: ECG Wave Delineation Using KNN*, IEEE, 2013, [Online], Available: https://ieeexplore.ieee.org/document/6614392

[24] Z. Zhao, L. Yang, D. Chen, Y. Luo, *A Human ECG Identification System Based on Ensemble Empirical Mode Decomposition*, in Sensors, vol 13, issue 5, 2013, pages 6832-6864, [Online], Available: https://doi.org/10.3390/s130506832

[25] MathWorks, 2018, *Signal Classification Using Wavelet-Based Features and Support Vector Machines*, Accessed: April 2021, [Online]. Available: https://au.mathworks.com/help/wavelet/ug/ecg-classification-using-wavelet-features.html

[26] T. Wang, C. Lu, Y. Sun, M. Yang, C. Liu, C. Ou, *Automatic ECG Classification Using Continuous Wavelet Transform and Convolutional Neural Network*; in Entropy, vol. 23, issue 1, 2021; [Online]. Available: https://www.mdpi.com/1099-4300/23/1/119/htm

[27] T.W. Bae, K.K. Kwon, *Efficient Real-Time R and QRS Detection Method Using a Pair of Derivative Filters and Max Filter for Portable ECG Device*, in Applied Sciences, vol. 9, issue 19, 2019; [Online], Available: https://www.mdpi.com/2076-3417/9/19/4128/htm

[28] I. Saini, D. Singh, A. Khosla, *QRS Detection Using K-Nearest Neighbour Algorithm (KNN) and Evaluation on Standard ECG Databases*, in Journal of Advanced Research, vol. 4, issue 4, 2013, pp 331-344; [Online], Available: https://www.sciencedirect.com/science/article/pii/S209012321200046X

[29] S. Saadatnejad, M. Oveisi, M. Hashemi, *LSTM-Based ECG Classification for Continuous Monitoring on Personal Wearable Devices*, in IEEE Journal of Biomedical and Health Informatics, vol. 24, issue 2, 2020, [Online], DOI: 10.1109/JBHI.2019.2911367

[30] M. Rashed-Al-Mahfuz, M.A. Moni, P. Lio, S.M.S. Islam, S. Berkovsky, M. Khushi, J.M.W. Quinn, *Deep Convolutional Neural Networks Based ECG Beats Classification to Diagnose Cardiovascular Conditions*, in Biomedical Engineering Letters, vol 11, 2021, pp 147-162; [Online], Available: https://link.springer.com/article/10.1007/s13534-021-00185-w

[31] A.M. Alqudah, *Towards classifying non-segmented heart sound records using instantaneous frequency-based features*, in Journal of Medical Engineering and Technology, vol. 43, issue 7, pp. 418-430, 2019; Available: https://doi.org/10.1080/03091902.2019.1688408

[32] M. Dorraki, A. Fouladzadeh, A. Allison, B.R. Davis, and D. Abbott, *On moment of velocity for signal analysis*, in Royal Society Open Science, vol. 6, 2019; [Online], Available: http://dx.doi.org/10.1098/rsos.182001

[33] M. Dorraki, M.S. Islam, A. Allison, and D. Abbott, *Parameter identification using moment of velocity*, in Royal Society Open Science, vol. 6, 2019, [Online], Available: http://dx.doi.org/10.1098/rsos.190671

[34] N. Emanet, *ECG Beat Classification by Using Discrete Wavelet Transform and Random Forest Algorithm*, IEEE, 2009, [Online]. DOI: 10.1109/ICSCCW.2009.5379457

[35] H. Li, et al., *Arrhythmia Classification Algorithm Based on Multi-Feature and Multi-Type Optimised SVM*, in the American Scientific Research Journal for Engineering, Technology and Sciences (ASRJETS), vol. 63, No 1, 2020, pp 72-86; [Online]. Available: https://asrjetsjournal.org/index.php/American_Scientific_Journal/article/view/5509/2046

[36] Y. Zhang, S. Wei, L. Zhang, C. Liu, *Comparing the Performance of Random Forest, SVM and Their Variants for ECG Quality Assessment Combined with Nonlinear Features*, in Journal of Medical and Biological Engineering, vol. 39, 2019, pp 381-392. [Online], Available: https://link.springer.com/article/10.1007/s40846-018-0411-0

[37] Z. Dokur, T. Olmez, *Heartbeat Classification by using a Convolutional Neural Network Trained with Walsh Functions*, in Neural Computing and Applications, vol. 32, 2020, pp 12515-12534; [Online]. Available: https://link.springer.com/article/10.1007%2Fs00521-020-04709-w

[38] O.S. Lih, et al., *Comprehensive Electrocardiographic Diagnosis Based on Deep Learning*, in Artificial Intelligence in Medicine, vol. 103, 2020; [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0933365719309030

[39] B. Hou, J. Yang, P. Wang, R. Yan, *LSTM-Based Auto-Encoder Model for ECG Arrythmias Classification*, in IEEE Transactions on Instrumentation and Measurement, vol. 69, issue 4, 2020, [Online], DOI: 10.1109/TIM.2019.2910342

[40] O. Yildirim, *A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification*, in Computers in Biology and Medicine, vol. 96, pp 189-202, 2018, [Online], Available: https://doi.org/10.1016/j.compbiomed.2018.03.016

[41] F. Bouaziz, D. Boutana, H. Oulhadj, *Diagnostic of ECG Arrhythmia using Wavelet Analysis and K-Nearest Neighbour Algorithm*, IEEE, 2018; [Online]. Available: https://ieeexplore.ieee.org/document/8652020

[42] O. Castillo, P. Melin, E. Ramirez, J. Soria, *Hybrid Intelligent System for Cardiac Arrhythmia Classification with Fuzzy K-Nearest Neighbours and Neural Networks Combined with a Fuzzy System*, in Expert Systems with Applications, vol. 39, issue 3, 2012, pp 2947-2955. [Online], Available: https://www.sciencedirect.com/science/article/pii/S0957417411012875

[43] R. Mahajan, R. Kamaleswaran, J.A. Howe, O. Akbilgic, *Cardiac Rhythm Classification from a Short Single Lead ECG Recording via Random Forest*, in 2017 Computing in Cardiology, 2017, [Online], DOI: 10.22489/CinC.2017.179-403

[44] M. Kropf, D. Hayn, G. Schreier, *ECG Classification Based on Time and Frequency Domain Features using Random Forests*, in 2017 Computing in Cardiology, 2017, [Online], DOI: 10.22489/CinC.2017.168-168

[45] M. Dorraki, (2017), *Moment-of-Velocity*, code via GitHub, [Online], Available: https://github.com/Dorraki/Moment-of-Velocity/blob/master/MomentOfVelocityII.m

Appendix A. One-Page Review

(Copied exactly as-is, including the Reference List.)

Can we Teach a Machine to be a Cardiologist?

Medical equipment, such as electrocardiograms (ECG), play a pivotal role in the diagnosis of a patient. In particular they make it possible for medical professionals to determine heart abnormalities and administer the correct treatment [1]. Heart disease continues to be a leading cause of death [2], so identifying and treating these diseases early is critical.

ECGs measure the electrical activity of the heart, which is then plotted as a waveform. Any irregularity in the plotted waveform can be indicative of an abnormality [3], so they are a useful tool for medical professionals in assessing patient health. An example of an ECG signal, including relevant points and intervals and their definitions can be found in Figure 1 at the end of this document.

Classifying ECGs is a challenging process for a number of reasons. Namely, normal ECGs may differ between individuals, one disease may have dissimilar signs on different patients, and two distinct diseases may have a similar effect on a normal ECG [3].

Recently, there has been an interest in employing machine learning (ML) in the medical field [1] [2] [4], such as by analysing the features of ECG to detect abnormalities. ML techniques could make it possible to diagnose patients more precisely than when done manually [3].

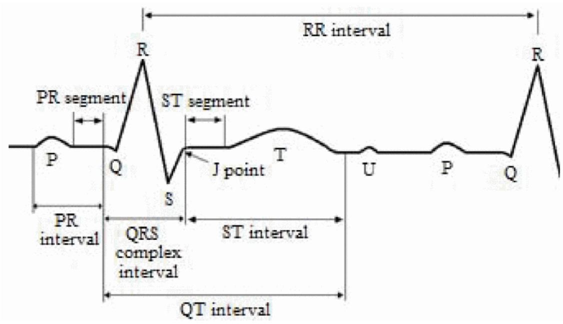
Prior to analysing the ECG for features, it is important to complete some pre-processing on the signal. This is done to remove baseline wander, motion artifacts, and other interruptions present in the collected results [5]. These noise removal techniques can be done with simple filtering techniques such as low pass filtering and Butterworth filters [6], or they can be based on adaptive filtering methods such as wavelet transforms, discrete Fourier transform (DFT) and the Pan-Tompkins algorithm [3] [5]. Adaptive filtering can produce much better results than the simple filtering methods.

Now it is possible to extract the relevant features of the ECG. In the time domain this may involve identifying the P and T waves, as well as the QRS complex over many cycles. It may also involve measuring the time between R peaks for consistency. In the frequency domain, a number of other features are worth identifying, including the very low frequency, low frequency and high frequency components of the signal [5].

From the features extracted, the signal can be classified as normal or abnormal. It may also be possible to determine the type of abnormality present, and further group the signal according to this. Various machine learning techniques have been found to be effective for this purpose. These include artificial neural networks (ANNs), the K-Nearest-Neighbour (KNN) Rule, Support Vector Machine (SVM) and decision tree classifiers [1] [3] [4] [5].

The SVM is a classification algorithm which involves using a hyperplane to provide an optimal decision boundary to separate classes [1], in this case normal and abnormal ECG signals. It can efficiently learn nonlinear functions and has been used previously in various pattern classification and regression applications [6].

One abnormality which can be detected from an ECG are heart murmurs. Heart murmurs are a “whooshing, humming or rasping” sound between heartbeats, and are caused by turbulent blood flow through the heart [7] [8]. Many are innocent, meaning they do not correspond to an underlying problem, but heart murmurs can also be linked to a range of disorders including congenital heart disorders, cardiac tissue damage and emotional stress [7]. They are often asymptomatic and are only picked up during routine health checks. Hence it is important that these are detected early, and treatment administered.



Feature	Description	Duration
RR	interval between R wave and the next R wave	0.6-1.2 s
P	first short upward movement of the ECG	80ms
PR	measured from the beginning of the P wave to the beginning of the QRS complex	120-200 ms
QRS	normally begins with a downward deflection Q, a larger upwards deflection R and ends with a downward S wave	80-120 ms
PR	connects the P wave and the QRS complex	50-120 ms
J-point	The point at which the QRS complex finishes and the ST segment begins is called J-point.	Not applicable
ST	connects the QRS complex and the T wave	80-120 ms
T	normally a modest upward waveform	160 ms
ST	measured from the J point to the end of the T wave	320 ms
QT	measured from the beginning of the QRS complex to the end of the T wave	420 ms
U	normally has low amplitude and often it is completely absent	Not mentioned

Figure 1: ECG signal, points of interest and their descriptions

References

[1] A. Ahamed, et al.; *ECG Heartbeat Classification Using Ensemble of Efficient Machine Learning Approaches on Imbalanced Datasets*; IEEE, 2020; Accessed 20 March 2021; [Online] DOI: 10.1109/ICAICT51780.2020.9333534

[2] K. Chen, A. Mudvari, F. G. G. Barrera, L. Cheng, and T. Ning; *Heart Murmurs Clustering Using Machine Learning*; IEEE, 2018; Accessed 25 March 2021; [Online] DOI: 10.1109/ICSP.2018.8652329

[3] S. H. Jambukia, V. K. Dabhi, H. B. Prajapati; *Classification of ECG signals using machine learning techniques: A survey*; IEEE, 2015; Accessed: 16 March 2021; [Online] DOI: 10.1109/ICACEA.2015.7164783

[4] A. D. Levin, A. Ragazzi, S. L. Szot, T. Ning; *A Machine Learning Approach to Heart Murmur Detection and Classification*; IEEE, 2020; Accessed 25 March 2021; [Online] DOI: 10.1109/CISP-BMEI51763.2020.9263528

[5] C. Venkatesan, et al.; *ECG Signal Preprocessing and SVM Classifier-Based Abnormality Detection in Remote Healthcare Applications*; IEEE, 2018; Accessed 20 March 2021; [Online] DOI: 10.1109/ACCESS.2018.2794346

[6] S. Celin, K. Vasanth; *ECG Signal Classification Using Various Machine Learning Techniques*; 2018; Accessed: 16 March 2021; [Online] Available: https://doi.org/10.1007/s10916-018-1083-6

[7] BetterHealth; *Heart Murmur*; Accessed 23 March 2021; [Online] Available: https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/heart-murmur

[8] L. Sherwood; *Human Physiology: From Cells to Systems*, 9th ed. Cengage Learning, 2016; ISBN: 879-1-285-86693-2